Dissertation

GPIODVS: General Purpose Intra-Operation Dynamic Voltage Scaling

Table of Contents

[Chapter 1: Introduction 3](#_Toc419742161)

[System Definition 3](#_Toc419742162)

[Linear / Low-Dropout Regulator (LDO) 3](#_Toc419742163)

[Charge Pump 4](#_Toc419742164)

[Switched Mode Power Supply (SMPS) 5](#_Toc419742165)

[Chapter 2: Related Work 6](#_Toc419742166)

[Dynamic Power Management 6](#_Toc419742167)

[Dynamic Voltage (and Frequency) Scaling 6](#_Toc419742168)

[Embedded Peripherals 6](#_Toc419742169)

[Chapter 3: Intra-Operation Dynamic Voltage Scaling 7](#_Toc419742170)

[Introduction 7](#_Toc419742171)

[Related Work 8](#_Toc419742172)

[Assumptions 9](#_Toc419742173)

[Methods and Materials 9](#_Toc419742174)

[Results 11](#_Toc419742175)

[Microchip MCP25AA512 EEPROM 11](#_Toc419742176)

[Numonyx M25PX16 Serial Flash 13](#_Toc419742177)

[Micro-SD Memory Card 15](#_Toc419742178)

[Honeywell HIH6130 Temperature / Humidity Sensor 20](#_Toc419742179)

# Chapter 1: Introduction

Embedded systems exchange dedicated functionality for efficiency and accuracy. These systems are designed to perform a specific set of functions typified by sensing, control and communications. Each of these tasks requires the expenditure of energy outside the predictable computational energy budget of the embedded processor.

For example, there is some energy expenditure necessary to communicate data from a microcontroller (MCU) to a peripheral device to which it is connected. The MCU may use a number of internal peripherals to communicate such as the Serial Peripheral Interface (SPI), Inter-Integrated Communications bus (I2C) or even General Purpose Input/Output pins (GPIO). The energy expenditure of MCU to device communication can be defined as:

is the power consumed by the processor with the transmitter enabled. is the power exhausted due to losses across the communications bus and is the power used by the external device to receive the transmission. Of course the total energy expenditure is the aggregate of power consumption throughout the communications interval.

Generally, the communications interval can be minimized by increasing slew rate of the transmission. The goal being to decrease the amount of time necessary to achieve the voltage necessary to register a high-level or “one”, . It is possible to increase the slew rate in two ways. The current sourcing capability of the transmitter can be increased, or the signaling voltage can be increased. Increasing the signaling voltage increases slew rate due to the capacitive nature of the communications bus governed by:

Figure 1 shows the effect of slew rate as it affects a theoretical maximum communications speed. Initially we consider a device with a low signaling voltage and a low current sourcing capability (high series equivalent series resistance). This device (Low V, Low I) takes the longest amount of time to reach the minimum. Increasing the source capability is effective, yet often impractical because to do so would necessitate larger semiconductors. Also, increasing the source capability tends to increase leakage currents and presents eliminates the intrinsic short-circuit protection afforded by current limited outputs. By doubling the source capacity, the slew rate is nearly doubled as well, the (Low V, High I) trace achieves the minimum voltage at 10ns.

A more practical approach is demonstrated with the (High V, Low I) trace. Many inputs tolerate a wide voltage range and minimum of 1.6 volts is typical of a device powered by a 3.3 volt supply [1]. As the example of Figure 1 shows, increasing the voltage can be a very effective way of increasing slew rate. The (High V, Low I) trace shows how the receiver can achieve the minimum input voltage at 11ns rather than 19ns. In fact, this method is widely followed and a large number of peripheral devices specify their communications speed as a function of their operating voltage.

Minimizing the time spent communicating is of the utmost importance because both the MCU and target device must remain in an active state throughout the transaction. After communication completes, both the MCU and target can typically return to sleep mode where power consumption is drastically reduced. Sleep functionality is common in most embedded devices and therefore the incremental increase in by increasing signaling voltage is more than offset by the decreased duration of the total power consumption.

Figure : Effects of Slew Rate on Theoretical Maximum Communications Speed

While it is established that communications speed is dependent on voltage, many devices have voltage-independent states. For example, a device may have varying communications performance throughout the range of 1.8V – 5.5V, but performs specific functions (sensing, controls, memory) identically throughout the voltage range. In fact this arrangement is common throughout the millions of available peripherals. The remainder of this paper investigates energy and delay optimization of specific device operations by minimizing the delay of voltage dependent states and minimizing the power consumption of voltage independent states.

This work focuses on extracting those efficiency gains by performing the following investigations:

1. System definition:
   1. The application microcontroller (MCU). With specific attention to dynamic voltage and frequency scaling (DVFS) and dynamic power management (DPM) capabilities.
   2. The characteristics of peripherals attached to the MCU. Fine-grained dynamic loading characteristics are of particular interest.
   3. The efficiency characteristics of one or more power supplies that provide power to the MCU and peripherals.
2. Intra-Operation Dynamic Voltage Scaling (IODVS)
   1. Fine-grained modulation of peripheral power supplies enables significant energy savings with no-effect on peripheral performance.
   2. Developing timing and energy based heuristics to maximize the benefit of IODVS
3. Supervised IODVS
   1. IODVS can interfere with the functionality of some peripherals on the same domain. Identifying interfering voltage ranges and quantifying their impact allows us to increase the overall efficiency of the system.
   2. A peripheral voltage supervisor addition to the uC/OS-III kernel that uses DPM heuristics to balance IODVS voltage changes against predicted break-even times.
4. In-system characterization
   1. Sweeping across the full range of input and output voltages and across a full range of output currents to determine a comprehensive efficiency graph for the application power supplies.
   2. Performing a complete set of peripheral operations while metering current consumption and operation duration across a full range of operating voltages.
5. Peak-efficiency scheduling
   1. Peripheral power characterization and real-time scheduling techniques are used to determine optimal strategies for maximizing power supply efficiency
   2. The technique is extended to systems with power supplies from which peripherals can select to operate.

We maximize switched-mode power supply (SMPS) efficiency throughout the system by paying special attention to the loading characteristics of embedded peripherals. We exploit adjustability of the power supply to minimize those loads. Furthermore, we explore techniques such as DVFS, DPM and real-time scheduling to function as a counterweight to peripheral loads in order to keep SMPS efficiency at the maximum. Finally, we extend the technique to systems that make use of multiple SMPS domains and explore switching peripheral loads to consume energy at their maximum efficiency.

Switched-mode power supply (SMPS) efficiency is maximized throughout the system by paying special attention to the loading characteristics of embedded peripherals. The adjustability of the power supply output voltage is exploited to minimize those loads. Furthermore, techniques such as DVFS, DPM and real-time scheduling are employed to function as a counterweight to peripheral loads in order to keep SMPS efficiency at the maximum. Finally, we extend the technique to systems that make use of multiple SMPS domains and explore switching peripheral loads to consume energy at their maximum efficiency.

# Chapter 2: System Definition

## Linear / Low-Dropout Regulator (LDO)

Voltage translation is a basic necessity for most embedded systems. The majority of embedded systems are supplied a voltage that is higher than is required to operate. Often this is due to legacy requirements as embedded systems trend toward lower operating voltages. However, on a more practical level, the higher supply voltage provides margin that allows for a certain amount of voltage droop to be tolerated by end-devices. There are three common methods of DC-DC conversion in order to accomplish the step-down.

The least complex circuit for step-down applications is the linear regulator shown in Figure 1. The circuit requires that the input voltage be maintained at some level higher than the output voltage. This margin is known as the dropout voltage or Vdropout. Modern versions of the circuit have focused on decreasing this margin and are known as LDOs (Low Dropout Regulators).

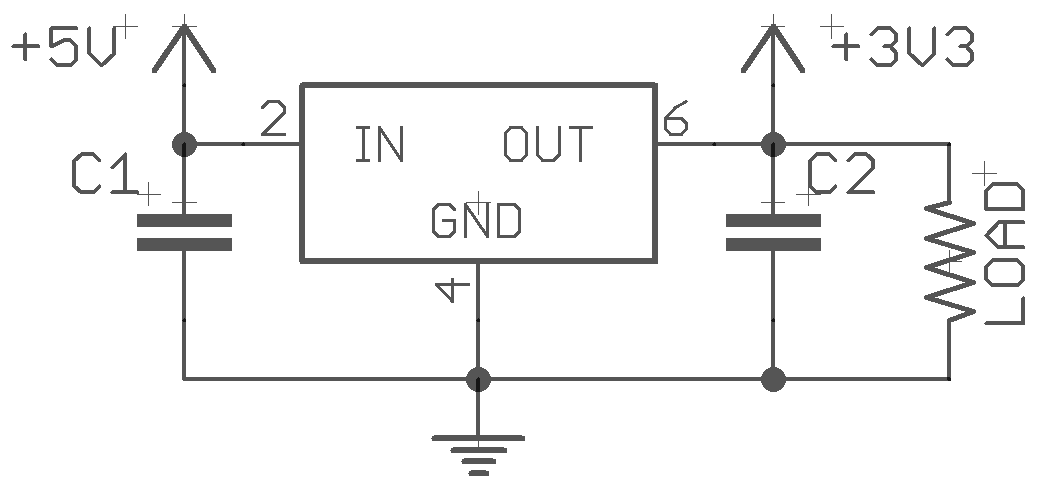


Figure : A Linear Regulator / LDO Circuit

This application requires the fewest external components thus minimizing cost and PCB area. It also produces the least amount of noise on the load side of the circuit. However it is the least efficient at DC-DC conversion. The power consumed by the linear regulator is modeled by considering both the conversion and quiescent loads: . The quiescent current is usually so low as to be ignored and thus the efficiency of the converter can be approximated to . Thus the regulator is unsuitable for translating large voltage differentials. Linear regulators tend to dissipate large amounts of heat due to their inefficiency and thermal limitations often limit their applicability.

## Charge Pump

Another method of DC-DC conversion can be accomplished via the charge pump. This circuit has the benefit of not requiring a physically large inductor and still provides more efficient translation of large voltage differentials than the LDO. Figure 2 shows a typical application and they generally require only a few external capacitors in order to function. Additionally, they are capable of generating DC voltages below the ground level of the input. They are commonly found in TTL🡪RS232 converters because RS232 signaling has a very wide voltage range (typically +-13V on modern implementations).

The current driving capacity of the charge pump is limited by both the size of the external capacitors and by the switching frequency of the device. The efficiency of the device is dictated by many factors, but the typical charge pump will be ~15% less efficient than a Buck regulator across the current output range. Taking all of these factors into account, they are best suited for translating a wide input voltage range into a potentially wide output voltage range and at low current.

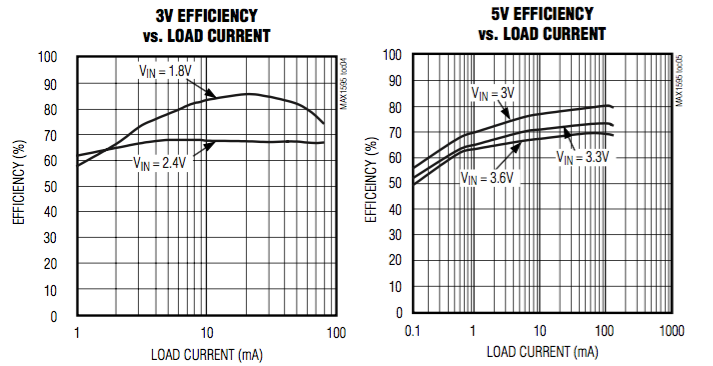
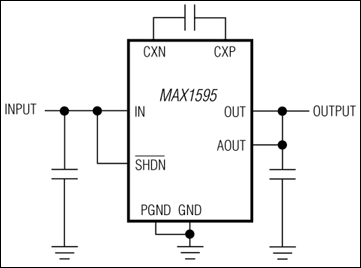


Figure : A Typical Charge Pump Circuit and Efficiencies [1]

## Switched Mode Power Supply (SMPS)

A common application of SMPS in embedded systems is DC-DC conversion. The converter is tasked with translating from one voltage level to another. For the majority of embedded systems this responsibility is to step down voltage from a high level to a lower level in a buck configuration as shown in Figure 1:

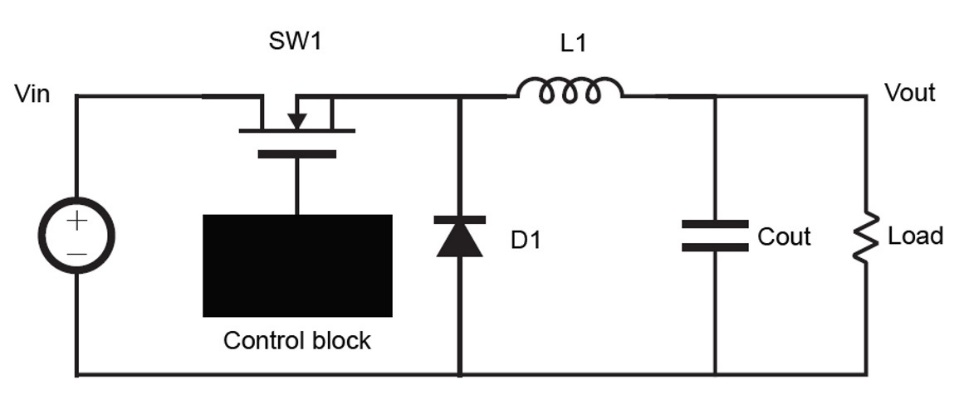
[

Figure : A Simple SMPS in Non-Synchronous Buck Configuration [1]

The SMPS has a number of advantages over charge pumps and linear regulators. The SMPS can translate large voltage differentials with high efficiency. The efficiency of the converter is not related to the input to output voltage differential. Rather, the efficiency is dominated a combination of conduction losses and switching losses. Conduction losses are the I2C losses incurred due to current flow through the inductor and transistor. Switching losses are incurred by charging and discharging the gate of the MOSFET. Thus, high switching frequencies will cause high switching losses.

# Chapter 2: Related Work

Embedded power management to date is split into two distinct fields: Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS).

## Dynamic Power Management

DPM techniques exploit power switching capabilities (clock-gating for example) in order to disable sections of the system while they are unused. Of course, disabling the section entirely results in a wake-up time for that section and therefore, significant research has gone into determining the optimal time to wake up the disabled section so as to have no increase in latency.

The break-even time is the figure of merit for DPM as it pertains to energy savings. If a device requires a long wake-up time (a time in which it is incapable of being used), then it is incumbent on the system to determine how to best schedule the disabling of the device.

## Dynamic Voltage (and Frequency) Scaling

Modern embedded Systems on Chip (microcontrollers) consume power at a rate of where f is the switching frequency of the circuit and C is the MOSFET gate capacitance. Vdd, the switching voltage is ripe for optimization because power consumption is proportional to the square.

In addition to the substantial power savings afforded by decreasing Vdd, microcontrollers also have a linear relationship between maximum possible switching frequency and the switching voltage. Thus, reductions in voltage result in decreasing the maximum possible frequency of the microcontroller.

In situations where peak performance is not required of the system, the frequency is adjusted downwards simply to minimize the linear part of the power consumption equation. Thus, if possible, it is also desirable to adjust the switching voltage to match the switching frequency.

## Embedded Peripherals

Most research to date regarding the energy optimization of embedded peripherals makes use of DPM. This is natural because most embedded peripherals include some form of ‘shutdown’ mode that allows the system to drastically decrease the static power consumption of the device. Thus, a significant amount of research has gone into determining the optimal breakeven time of embedded peripherals.

No research to date has investigated the voltage / performance relationship of embedded peripherals.

# Chapter 3: Intra-Operation Dynamic Voltage Scaling

## Introduction

Consider an embedded system where the supply voltage to an application MCU is decoupled from the supply voltage of its peripherals as shown in Fig. 1. This design is becoming more common as modern MCU applications take advantage of Dynamic Voltage and Frequency Scaling (DVFS) and, in effect, IODVS is a natural extension of DVFS to the peripheral domain. We find that energy can be saved by lowering the peripheral domain voltage during periods where low-performance is allowed such as mandatory wait periods and where the application of traditional DVS or DPM techniques would adversely affect operation of either the device or the system.



1. An IODVS Enabled System

For example, EEPROM is a typical peripheral device that is used to provide non-volatile data storage. They are usually specified for use in systems that require a quick data access time and have low storage capacity requirements. The chips are often specified to operate at multiple voltage levels to achieve compatibility with systems using voltages from 1.8V to 5.0V.

A write operation to the SPI device (and optional verification stage) is typified by the timing diagram shown in Fig. 2. Maximum communication speed scales with slew rate and therefore scales with voltage. It follows that communication between the MCU and peripheral domains should occur at matched voltages, thereby maximizing data transfer, minimizing energy delay product (EDP) and eliminating the need for voltage level translation.



1. A SPI EEPROM Write / Verify Cycle

The most distinct benefit of IODVS can be realized during the longest portion of the transaction described in Fig. 2: the delay. IODVS decreases the supply voltage to the chip during this voltage-independent period and we find that the energy cost of the transaction is significantly decreased.

The IODVS technique is applicable to any peripheral that has a voltage/frequency dependence and particularly applicable to those with a wait-state. Our investigation considered the peripherals listed in Table I as a representative sample. The device descriptions and voltage requirements are listed next to their physical location on the test fixture.

Enabling IODVS requires only an adjustable power supply and a means of modulating the output voltage. A switched mode power supply (SMPS) is preferable because it is an efficient means of translating voltage levels. An adjustable linear regulator could be used, but only the benefits of decreased current consumption would be realized.

1. Typical External Peripherals

|  |  |  |
| --- | --- | --- |
|  | Honeywell HIH-6130 I2C  Temperature / Humidity Sensor | Vmax: 5.5V  Vmin: 2.3V |
| Microchip MCP 25AA512  512Kbit (64KB) SPI EEPROM | Vmax: 5.5V  Vmin: 1.8V |
| Numonyx M25PX16  16Mbit (2MB) SPI Serial Flash | Vmax: 3.6V  Vmin: 2.3V |
| SPI Mode SD Cards:  Lexar SDSC: 1.0GB  Sandisk SDSC 1.0GB  SwissBit: SDSC 512MB  Kingston: SDSC: 2.0GB | Vmax: 3.6V  Vmin: 2.7V (Operating)  Vmin: 2.0V (Idle/Ready) |

IODVS was tested on each of the seven sample peripherals listed in Table I. Each device was characterized by a peripheral power profile (PPP) that is effectively a lookup table of state-voltage pairs. The state voltages were derived from the specifications of the device datasheet. A common sequence of operations was performed repeatedly and random input parameters were used on each iteration. The output was analyzed, and no failures or unexpected behavior occurred. We observed idle energy decreases of up to 66% and intra-operational energy decreases of up to 40%.

## Related Work

Dynamic Power Management and Dynamic Voltage Scaling implementations seek to minimize energy consumption in embedded systems. DPM policies tend to focus on strict power-state relationships [1], while DVS policies tend to incorporate a linear power-performance relationship [2]. In fact, DVFS is so useful that hardware is designed specifically to take advantage of it [3]. Most DPM implementations focus on optimal scheduling techniques such that peripherals emerge from low-power states just in time for access by tasks. Generally, the approaches to date can be categorized as a combination of either online [4] or offline [5] and deterministic [6] or probabilistic [7].

Offline analysis can aid in the implementation of DPM by analyzing the control flow graph of an individual task or task set to determine when a peripheral is likely to be accessed [5]. Similar data can be realized online by profiling tasks and determining which paths lead to a peripheral access [8]. Both methods enhance the accuracy of predictions regarding the optimal peripheral wakeup time. In fact, all methods must evaluate the cost/benefit of peripheral deactivation with respect to the energy savings gleaned versus the time spent reactivating the device when next needed. This equality is commonly known as the breakeven time [9].

Peripherals can be operated under the same linear [10] and step-wise [11] constraints. Approaches have been explored with respect to optimally scheduling devices with multiple power saving states and with systems where multiple tasks share a common resource (inter-task DPM). Naturally, the decrease in voltage margin along with the decrease in available task slack time also decreases the ability to detect and correct errors as they occur [12]. In fact, DVFS is so useful that dual-output circuits have been developed primarily targeting systems with a SoC [13]. The same circuitry could be reused to implement IODVS.

IODVS is most similar to the Component Aware DVS technique [14] [15] developed for use in wireless sensor nodes (WSN) [16] [17]. An adjustable regulator is used to operate an embedded system at its minimum voltage requirements. However, CADVS operates at the task-level and therefore results in a power / performance relationship typical of DVS. IODVS differs in that it extends the technique into intra-operation granularity.

DPM techniques inherently interfere with the operation of the device and impose a lag in response time. IODVS instead exploits the acceptable operating voltages of the device and does so with no effect on response time.

## Assumptions

The MCU and peripheral voltage domains are decoupled and the peripheral domain is adjustable using an MCU-controlled DAC. All digital signaling between the MCU and the peripheral domain are made at the same voltage. The increased cost and decreased performance of level translation or isolation is too great to warrant implementation [18] for this purpose in most embedded systems. Above all, the lowest communication energy-delay product is found at matched voltage/frequencies.

The current measurements are taken at the output of the peripheral power supply. Thus, the data will indicate the effect of IODVS on the set of peripherals on the domain and not on any one peripheral in particular.

The PPP state-voltage lookup table of each device is constructed solely from the acceptable usage specifications contained within the device datasheet. It was discovered experimentally that many of the devices that we tested operated well below their specified minimum voltage requirements. Although minimizing energy consumption by means of minimizing voltage is the primary goal of this work, it is necessary to ensure functionality of the device is maintained across all environments that may degrade performance.

For instance, the EEPROM under test is specified to operate in the range of -40℃ to +80℃ and with a minimum endurance of 1,000,000 write cycles. As the device nears the edge of its acceptable operating temperature or approaches its lifetime write-cycle limit, the minimum necessary voltage to guarantee completion of a write operation is likely to be that specified by the designers along with an acceptable factor of safety.

## Methods and Materials

The TPS62240 [19] adjustable (SMPS) was selected to power the peripheral domain because of its high efficiency at light loads, output capacity and adjustability. Peripheral domain voltage modulation is accomplished using a DAC output on a STM32F205 MCU signaling into the resistive feedback circuit on the SMPS. To measure the results of IODVS, the domain is outfitted with current sense circuitry [20] on both the input to the SMPS and the output to the domain.



1. Peripheral Domain SMPS, Control and Current Sense Circuitry

As shown in Fig. 3, the adjustable peripheral power supply is outfitted with current sense resistors and amplifiers on both the input to the supply and the output to the peripheral domain. These signals, along with the input voltage to the supply and the output voltage to the domain are fed into the ADC of the STM32F205 microcontroller and sampled at up to 1MSPS. The MCU has 3 simultaneously sampling ADCs which allows for simultaneous measurement of the output voltage, input current and output current.

Peripheral operations are broken up into states per an intrinsic state transition diagram. For example, to perform a write to EEPROM, the MCU must issue the write command, write the data, wait for a specified delay period and then read the data back in order to verify a correct write. Therefore, the states are delineated as Idle, Writing, Waiting and Verifying.

Each peripheral operation is associated with a specific voltage. For instance, per our assumptions, data transfers must occur at equal voltages between the domain and MCU. The voltages of the writing and verifying states must then equal that of the MCU (3.3V). This leaves the idle and wait states available for voltage scaling.

For each device, the pairs of states to voltages form a lookup table (PPP). Each test designates a power profile to use. Peripheral memory was tested with random data and across random memory addresses. Tests were run 2048 times, and the results were averaged. While operating IODVS within the specifications of the device datasheet, no operations failed.

All test results were measured entirely in-system using the three 12-bit simultaneously sampling ADC converters onboard the MCU. The converters are triggered from a timer overflow using a reload value that allows for a complete buffer fill roughly corresponding to the expected length of the test. For example, the duration of the EEPROM test was approximately 10ms with a buffer size of 10240 samples yielded 976.6ns per sample (or a rate of 1.024MHz). Upon a trigger, the state of the peripheral is stored synchronously with the sample. Each test data set was retrieved upon completion and is composed of:

* Time Scale
* 10240 12-bit ADC Samples per channel
* Output Voltage
* Input and Output Current
* 10240 Device State Samples (reading / writing / etc.)
* Bit Resolution (ADC value 🡪 Current or Voltage)

The energy consumed throughout a test is calculated using the fundamental relationship shown in (1). The results were calculated offline via (2) and (3), where S is the state of the device, and Ts is the sampling period.

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |
|  | (3) |

Separating the energy consumption by state is important because it allows us to consider the effect of duty cycle on the results of IODVS. Each device has an idle state where the voltage applied to the device is the minimum allowed by specification. Applying IODVS to the idle state significantly decreases energy consumption while avoiding the increased response time of DPM.

Likewise, IODVS is applicable to an exploitable sequence of active operations, resulting in decreased energy consumption without the performance impacts of DVFS. We can separate energy consumption into two intervals as shown in (4).

|  |  |
| --- | --- |
|  | (4) |
|  | (5) |

A duty cycle of 0% will be dominated by and energy consumption will converge on that of the idle state. On the other hand, a duty cycle of 100% will be dominated by In which case, energy consumption converges on the weighted average of the set of states comprising the active period. In any case, the actual energy decrease because of IODVS will lie in between these two extremes.

## Results

### Microchip MCP25AA512 EEPROM

IODVS uses peripheral power profiles to correlate peripheral voltages with internal state. The PPP specified for the EEPROM under test is derived from the specifications of its datasheet [21]. The EEPROM can communicate at 10MHz at 3.3V, while only 1.8V is required for basic operation. However, the length of the mandatory page-write delay is voltage independent and exploitable by IODVS.

The standard PPP is considered a control group and mandates that all states (writing/waiting/verifying/etc.) should have 3.3V applied to the peripheral. The 1.8VIW (1.8V Idle/Wait) profile mandates that the EEPROM should have 1.8V applied during the idle and waiting states and 3.3V applied on all others. Fig. 5 provides a comparison of both the standard PPP and the 1.8VIW profiles enabled by IODVS.

The state transition diagram of Fig. 4 is known a-priori and is followed throughout the tests illustrated in Fig. 5. The black line indicates device state and is sampled synchronously with the voltage and current measurements.

The test begins with the EEPROM powered up and in the idle state. The WREN (write enable) command is transmitted to the peripheral, along with the write command and address which is followed by 128 bytes of random data (1 page-size). The peripheral and device driver then transition into the page-write delay state and the peripheral voltage is decreased to 1.8V. After the delay, the device driver increases the voltage to the 3.3V necessary for communication and then reads the data back from the device to verify that it was committed properly.

The effects of IODVS are most distinct during the Idle and Wait states. Energy consumption during these states decreased 66.7% and 48.7% respectively. Energy consumption during the Write state increased by 30%. This is primarily as a result of the energy required to charge the domain to 3.3V which is required to complete the transaction.

Note that although the current measurement appears to exceed the graph in Fig. 5, the current spike was indeed measured to be approximately 15mA and the data were integrated accordingly. In fact, charging the domain voltage is responsible for the 29% and 37% increases in the write and verify states respectively.

Two of the SPI lines on the test fixture are multiplexed for I2C communication. This causes the 1mA current swings during the communication phases of the test. The current consumption of the device indicates the behavior of the operation within. Two distinct periods of increased power demand are noticeable, these are likely to be an internal erase operation followed by a write.

The idle time of the test lasted 1ms out of a total test time of 9.475ms. The duty cycle of this test was 89.45%, and energy consumption was reduced by 26.67%. Removing the idle time from the total would yield an energy decrease of 22.36% at a duty cycle of 100%. Realistically, this type of device is used much less frequently owing to its finite number of useable write-cycles. At a duty cycle of 0%, the savings would converge on 66.7%.





1. EEPROM Energy Consumption

|  |  |  |  |
| --- | --- | --- | --- |
| State | Static (uJ) | IODVS (uJ) | Delta |
| Idle | 9.84 | 3.28 | -66.70% |
| Write | 13.28 | 17.08 | 28.61% |
| Wait | 62.03 | 31.83 | -48.69% |
| Verify | 16.12 | 22.08 | 36.96% |
| **Test Total** | **101.27** | **74.26** | **-26.67%** |

1. Energy Consumption vs. Duty Cycle

|  |  |  |  |
| --- | --- | --- | --- |
| **Duty Cycle** | Static avg. (uJ) | IODVS avg. (uJ) | Delta |
| **Duty: 0%** | 9.84 | 3.28 | -66.70% |
| **Duty: 25%** | 30.24 | 20.20 | -33.19% |
| **Duty: 50%** | 50.63 | 37.13 | -26.67% |
| **Duty: 75%** | 71.03 | 54.05 | -23.90% |
| **Duty: 100%** | 91.42 | 70.98 | -22.36% |

### Numonyx M25PX16 Serial Flash

Serial flash modules have a somewhat more complicated state transition diagram than EEPROM. Serial flash chips can only program zeroes to their memory locations. At a simplistic level, this requirement necessitates a complete erase of a subsector before modifying the memory within it. The M25PX16 [22] supports a minimum of 4KB (sub-sector) erase and a maximum of 256B (page) sequential writes. To perform a read-modify-write operation, the transition diagram shown in Fig. 6 is followed.

As with all of the devices under test, the control PPP was standardized at 3.3V throughout, while the 23VIW (2.3V idle/wait) PPP was constructed from the parameters listed within the datasheet. The device has a minimum operational voltage of 2.3V which is used for the idle and wait states. The subsector erase is specified to take a maximum of 150ms, while the page-write completes with a maximum delay of 5ms. Cross-subsector writes were not evaluated because that would simply require two test sequences to occur sequentially.

The device can reach the idle state either 10ms after power up or approximately 30us after the execution of a wake command. As the test begins, the chip is in the idle state; it does not require an initialization routine to execute before entering a functional state. A random sub-sector of memory is read into cache and is modified with the 128 bytes of random data to be committed. The sub-sector erase operation is executed, and IODVS adjusts the peripheral voltage to the wait state (2.3V in this PPP).

Upon completion of the erase cycle, the modified cached data are written back to the flash module one page at a time, resulting in 16 total page-writes. The writes cause a series of alternating “write-wait” states, and the corresponding voltage/state changes are evident in Fig. 7. After the final page-write delay is complete, the data are read back and verified with the cached copy to ensure data integrity.

Table IV summarizes the energy decrease per state yielded through the use of IODVS. As expected, the most significant savings are found in the idle and wait states, while an increase is seen in the active states.

Because the test was in the idle state for 1ms out of the total length of 257ms, the test represents a duty cycle of 99.6%, which is effectively the worst case. As the duty cycle decreases, the idle energy savings begins to dominate and pushes the average toward a limit of 48.66%.

It is noteworthy that this erase-write sequence is common to all flash memory, and so IODVS is applicable to flash memory in general. In high performance parallel NOR and NAND devices, writes complete on the order of micro-seconds. However, erase operations complete on the order of seconds and are easily exploitable by IODVS.

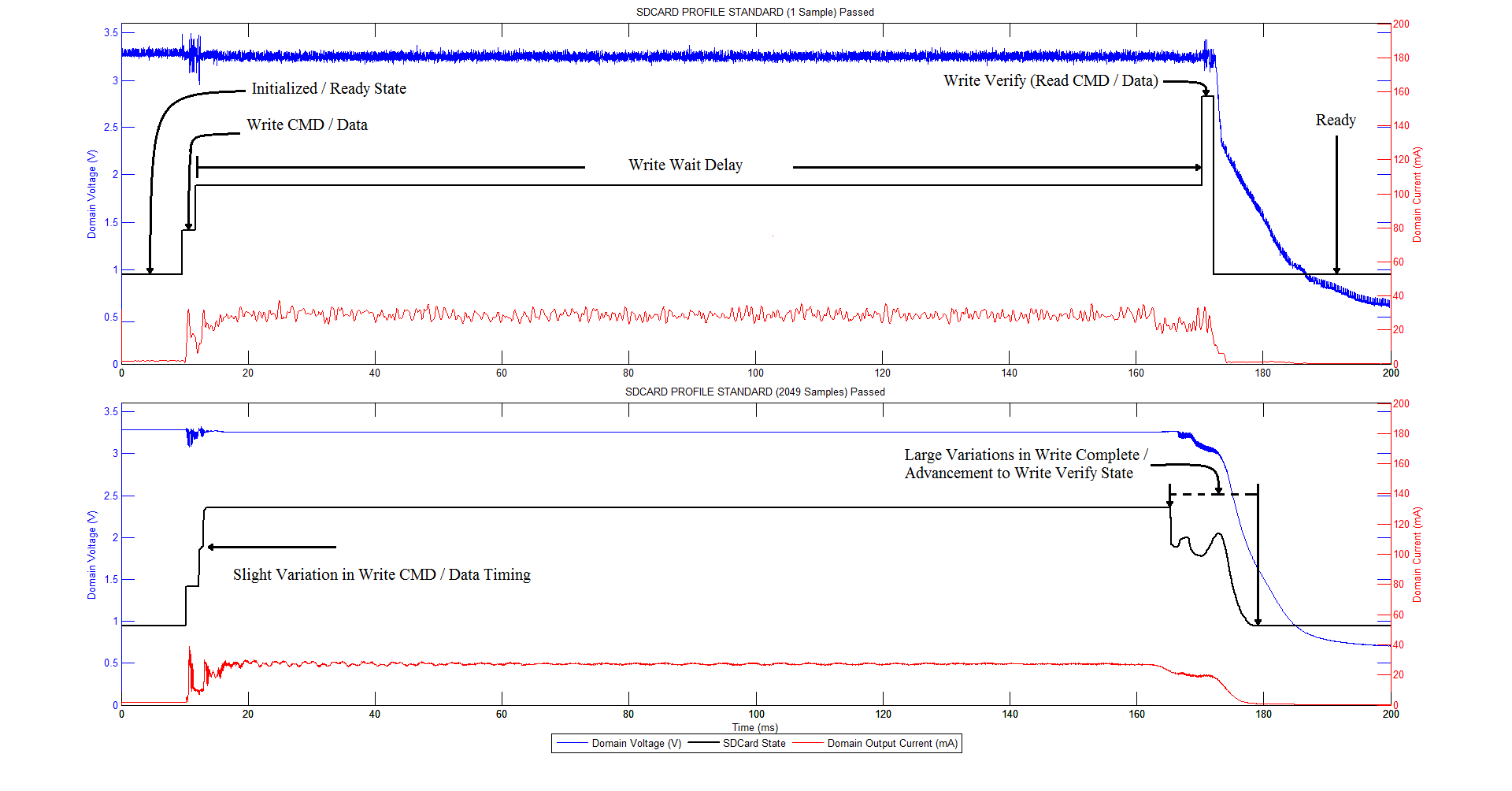




1. Serial Flash Energy Consumption

|  |  |  |  |
| --- | --- | --- | --- |
| State | Static (uJ) | IODVS (uJ) | Delta |
| Idle | 10.27 | 5.27 | -48.66% |
| Reading | 89.85 | 90.86 | 1.13% |
| Write\* | 80.35 | 89.20 | 11.02% |
| Wait\* | 551.18 | 344.92 | -37.42% |
| Verify | 57.52 | 72.45 | 25.96% |
| **Test Total** | **2517.04** | **1530.27** | **-39.20%** |
| **\***sequential write and wait states combined | | | |

### Micro-SD Memory Card





Micro-SD Cards follow a standard outlined by the SD Association [23]. The standard is a minimum set of electrical and communication specifications that must be met and some vendors exceed those specifications [24] [25] [26]. A few of the variable parameters include clock speed, slew rate, initialization time, block length, read/write timing and power consumption. Additionally, the devices use a MMU which causes access timing to vary. The cumulative effect of these variations results in the SD Card protocol relying heavily on device polling. We avoid polling during write operations by predicting the write completion time.

Fig. 8 shows how timing variations affected our testing. The top figure shows one write/verify operation with constant polling for write-complete. The bottom figure shows the average of 2049 operations with polling beginning at 165ms. The non-monotonicity of the device state from the 165ms to 180ms marks indicate that some portion of the writes completed before 165ms had elapsed and advanced to the verify state immediately after polling. The shape also indicates that the majority of writes completed and advanced around the 170ms mark. Tuning the optimal polling time is a topic for further research.

An SD Card must be initialized after power up as shown in Fig. 9. The MCU communicates with the SD Card via SPI and the initialization process typically takes 250ms. Not all SD Cards support power down modes. IODVS enables the device to transition to the 2.7V “Initialized” state, rather than undergoing a complete power-cycle and incurring the 250ms penalty as would be typical with DPM.

From the initialized state, the device was sent a write command to a random address with random data. The device driver then waits a predetermined amount of time (the prediction) before beginning to poll the device for write complete which can take up to 250ms. After the write finishes, the device driver reads the data back in order to verify that it committed properly before returning to the idle state.

The SD Card has the highest current consumption of the devices tested and therefore requires a bulk capacitor at the load in order to ensure sufficient supply at the device. The point at which domain capacitance is detrimental to IODVS is dependent on the demands of the loads. Larger loads allow the domain to transition to lower voltages faster, while larger capacitances cause the domain to transition more slowly.

#### Sandisk SDSC 1.0GB Micro-SD Memory Card

Initial experiments with the Sandisk Micro-SD indicated that the majority of write operations completed approximately 150-170ms after they began. Based on this data and as shown in Fig. 10, the card was not polled until the test reached the 180ms mark (which is approximately 165ms after the write command completed successfully). After write-complete polling begins, we find that all of the writes had already completed and were eligible to transition into the verification stage.

Idle energy consumption dropped by 11.5% and the idle duration accounted for 10ms of the 184.1ms test, yielding a duty cycle of 94.6%. A duty cycle of 100% (constant write/verify) would yield an energy decrease of 27.54%. The write and verify stages of the test were relatively unchanged, though this could be due to insufficient resolution. Based on previous tests with higher resolution, charging the domain took between 5-10uJ and therefore is negligible compared to the total decrease of 3893uJ.



1. Sandisk Micro-SD Card Energy Consumption

|  |  |  |  |
| --- | --- | --- | --- |
| State | Static (uJ) | IODVS (uJ) | Delta |
| Idle | 157.07 | 138.95 | -11.54% |
| Write | 26.48 | 26.23 | -0.93% |
| Wait | 14021.97 | 10126.95 | -27.78% |
| Verify | 89.86 | 91.88 | 2.25% |
| **Test Total** | **14295.38** | **10384.02** | **-27.36%** |

#### Lexar SDSC 1.0GB Micro-SD Memory Card

The Lexar Micro-SD card had a higher average power draw and a different write-completion characteristic than the Sandisk Micro-SD Card. The majority of writes completed between 140-180ms after the test began. This result can also be inferred from the drop in current consumption beginning at the 140ms mark. Polling for the completion did not begin until 160ms after the test began.

Despite the higher current draw, the system still benefited from a decrease in wait state energy consumption by 4049 uJ. The duty cycle was the same as the Sandisk card at 94.6% yielding an energy decrease of 24.12%.

Both the Sandisk and Lexar cards are older technology (manufactured in 2007 and 2009 respectively) and when compared with other cards, show higher energy consumption and slower performance. Newer implementations are better in both categories.



1. Lexar Micro-SD Card Energy Consumption

|  |  |  |  |
| --- | --- | --- | --- |
| State | Static (uJ) | IODVS (uJ) | Delta |
| Idle | 124.09 | 102.41 | -17.47% |
| Write | 34.52 | 34.42 | -0.28% |
| Wait | 16608.43 | 12558.83 | -24.38% |
| Verify | 39.45 | 56.87 | 44.17% |
| **Test Total** | **16806.48** | **12752.54** | **-24.12%** |

#### Swissbit S-200U 512MB Micro-SD Memory Card

The SwissBit Micro-SD Card is unique in that it uses 4x 4KB buffers to cache reads and writes to the memory card in order to speed up transaction times. The method is effective in that the worst case test time for the SwissBit card is less than half the best case test time for the previous two cards.

The card is equipped with power-fail circuitry that flushes the buffers to non-volatile memory once a voltage threshold has been reached. This functionality is seen at the moment just before the 70ms mark where the peripheral voltage reaches approximately 2.5V coinciding with a current spike of approximately 9mA.

The write-completion time varies much more significantly than the other cards. Current consumption of the device shown in Fig. 12 indicates that writes begin completing at approximately the 35ms mark.



1. SwissBit Micro-SD Card Energy Consumption

|  |  |  |  |
| --- | --- | --- | --- |
| State | Static (uJ) | IODVS (uJ) | Delta |
| Idle | 66.25 | 43.53 | -34.30% |
| Write | 25.01 | 25.72 | 2.85% |
| Wait | 3726.20 | 2839.78 | -23.79% |
| Verify | 36.31 | 31.68 | -12.74% |
| **Test Total** | **3853.76** | **2940.71** | **-23.69%** |

#### Kingston SDHC 2.0GB Micro-SD Memory Card

The Kingston Micro-SD Card was manufactured in 2014. Initial experiments with the device indicated that writes completed nearly 20x faster than the models previously tested. Furthermore, the maximum wait state duration appeared to be slightly over 1ms with a very high current consumption throughout the state. The test used a 2us sample time.

The write operation appears as a staircase between the 4ms and 6ms mark indicating that the device was ready for the write to a random address immediately in most cases, but after a 1ms delay in others.

Despite the fast characteristics of the device, IODVS was able to decrease the idle energy consumption by 31.4% and the current consumption of the wait state by 20.46%. The device was idle for 4ms out of the total test time of 12ms, yielding a duty cycle of 67%. The energy costs of the write, wait and verify states are relatively close. If the duty cycle were increased to 100%, the energy decrease would converge on 7.45%.



1. Kingston Micro-SD Card Energy Consumption

|  |  |  |  |
| --- | --- | --- | --- |
| State | Static (uJ) | IODVS (uJ) | Delta |
| Idle | 24.63 | 16.89 | -31.40% |
| Write | 89.74 | 91.45 | 1.90% |
| Wait | 122.44 | 97.39 | -20.46% |
| Verify | 54.00 | 57.53 | 6.53% |
| **Test Total** | **290.81** | **263.26** | **-9.47%** |

### Honeywell HIH6130 Temperature / Humidity Sensor

The MCU communicates with the temperature and humidity sensor [27] via I2C. The interface communicates in an open-drain fashion and therefore logic-high levels are accomplished simply by changing the MCU pin direction from output-low to input. The I2C bus was pulled to match the voltage level of the domain and therefore, when the MCU is sending data to the peripheral, it is not necessary to match the voltage of the MCU and peripheral domain. However, when the MCU is retrieving data from the peripheral, the voltages must be matched in order to ensure that input logic-level requirements are satisfied on the MCU.

The primary benefit of IODVS in the case of this peripheral is that the rate of I2C communication is highly dependent on the magnitude of the pull-up resistors enabling it and the signaling voltage. By allowing the voltage to increase to 3.3V during the read, larger pull-up resistors can be used, thus decreasing static power dissipation while maintaining the same communication frequency.

Because the device operates using open-collector signaling, the PPP is slightly different. Again, the control PPP is 3.3V in all states, but the IODVS PPP is 2.5VIRyTW (idle / ready / transmitting / waiting). Transmitting is denoted as seen from the MCU perspective. So the profile effectively mandates that only when the MCU is receiving data from the peripheral should it raise the device voltage to an MCU compatible level.

The test begins in the Idle state as shown in Fig. 14 and the MCU issues a “Measure” command to the sensor. The peripheral takes up to 4ms to wake from sleep [28] and then transitions to the temperature measurement and humidity measurement states in sequence. There is a noticeable drop in current in Fig. 15 upon the completion of the measurement and the MCU begins to read the data soon afterward.

This peripheral automatically enters an internal sleep mode described in its data sheet which drops the current consumption when a measurement has completed but has not yet been read. IODVS functions separately and provides additional energy savings. The first state has slightly higher energy power consumption because the device does not have a known measurement available.

IODVS consistently yields approximately 38% energy savings that is nearly duty-cycle independent. The reading state was unaffected by the optimization because voltages are equal.





1. HIH-6130 Energy Consumption

|  |  |  |  |
| --- | --- | --- | --- |
| State | Static (uJ) | IODVS (uJ) | Delta |
| Idle | 10.28 | 6.28 | -38.87% |
| Command | 1.68 | 1.05 | -37.60% |
| Waiting | 399.07 | 245.89 | -38.38% |
| Reading | 4.30 | 4.42 | 2.62% |
| **Test Total** | **415.33** | **257.64** | **-37.97%** |