General Purpose
Intra-Operation Dynamic Voltage Scaling

Daniel Moore

North Carolina State University

Department of Electrical and Computer Engineering

November 18, 2015

TABLE OF CONTENTS

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. System-wide loading characteristics dictate both the capacity and the strength requirements of the energy supply. These requirements impact the physical system in aspects of size, weight and cost as well as performance characteristics such as lifetime and thermal performance.

The impact of energy consumption is felt throughout the embedded system and it is therefore important to minimize energy consumption wherever possible. Extensive research has been performed focusing on minimizing energy consumption of the main processor by means of matching power consumption to performance demands. This research expands the search for energy savings outside of the processor domain and toward the peripherals to which it is connected. Primary attention is focused on performing voltage scaling of peripheral devices during both voltage-dependent and voltage-independent states.

Voltage Dependent States

A situation in which performance is correlated with applied voltage is a voltage-dependent state. Intra-system communication is an example of a voltage-dependent state and there is some energy expenditure necessary to communicate data from a microcontroller (MCU) to an in-system peripheral device. 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:

$$E = (P_{MCU} + P_{Loss} + P_{DeviceRx}) * T_{Comms}$$

P_{MCU} is the power consumed by the processor with the transmitter enabled. P_{Loss} is the power exhausted due to I^2R losses across the communications bus and $P_{DeviceRx}$ 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.

The communications period T_{Comms} is fundamentally limited by the aggregate aperture time necessary to transmit a packet of data ($T_{aperture} * N_{bits}$). The aperture time is the summation of data setup and hold times as shown in Figure 1. The setup and hold times can be affected by a number of factors, but primarily they are minimized by decreasing the transition time across the threshold voltage V_t of the receiving circuit. The result directly effects the slope of the clock line in Figure 1 and extends or contracts the setup and hold times accordingly.

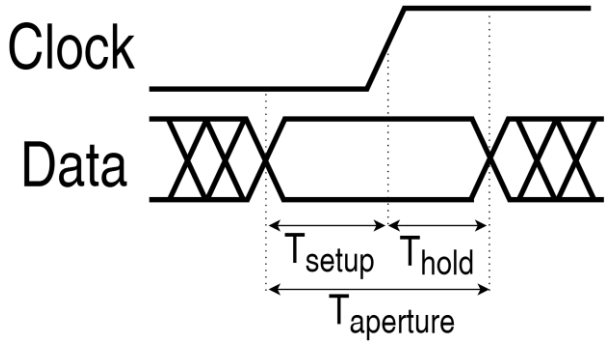


Figure 1: Aperture, Setup and Hold Times

The slope of these transitions is commonly known as the slew rate and the communications interval can therefore be minimized by increasing slew rate of the transmission. The goal being to decrease the amount of time required to achieve the voltage necessary to register a high-level or "one", V_{IH} for positive edges. Likewise, it is desirable to decrease the amount of time required to register a low-level or "zero" V_{IL} for negative edges.

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 physical communications link governed by:

$$V = V_{sig} * \left(1 - e^{\frac{-t}{RC}}\right)$$

Figure 2 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 V_{IH}. 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 noise while also eliminating 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 V_{IH} of 1.6 volts is typical of a device powered by a 3.3 volt supply [1]. As the example of Figure 2 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.

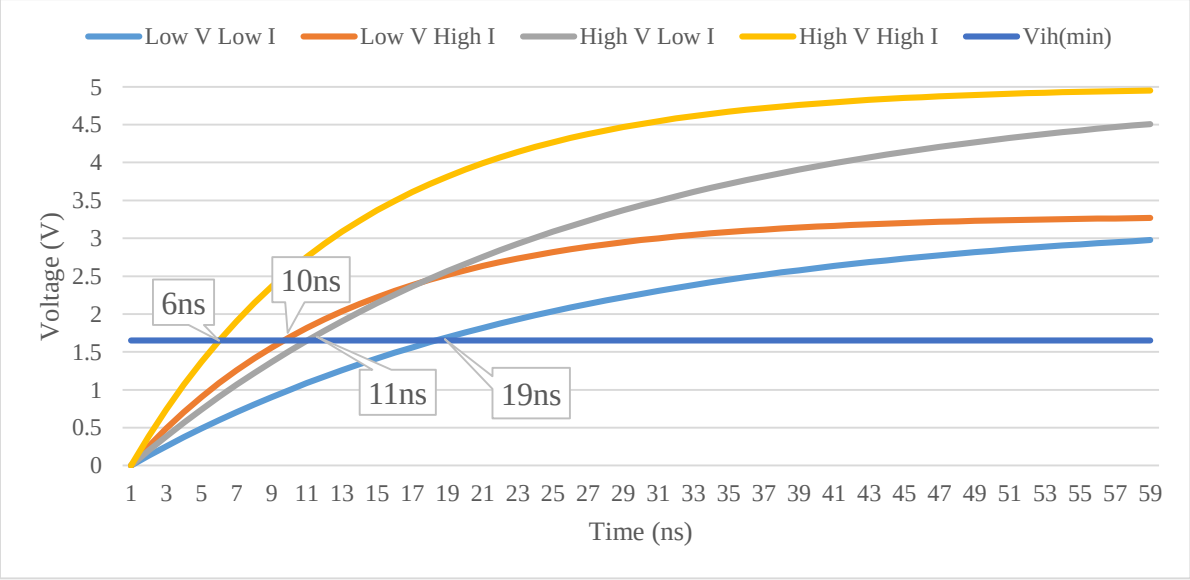


Figure 2: Effects of Slew Rate on Theoretical Maximum Communications Speed

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 P_{Loss} by increasing signaling voltage is more than offset by the decreased duration of the total power consumption.

Voltage Independent States

While it is established that overall communications speed is dependent on voltage, many devices have voltage-independent states where performance is not dependent on the supply voltage. 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 thousands of commercially available peripherals.

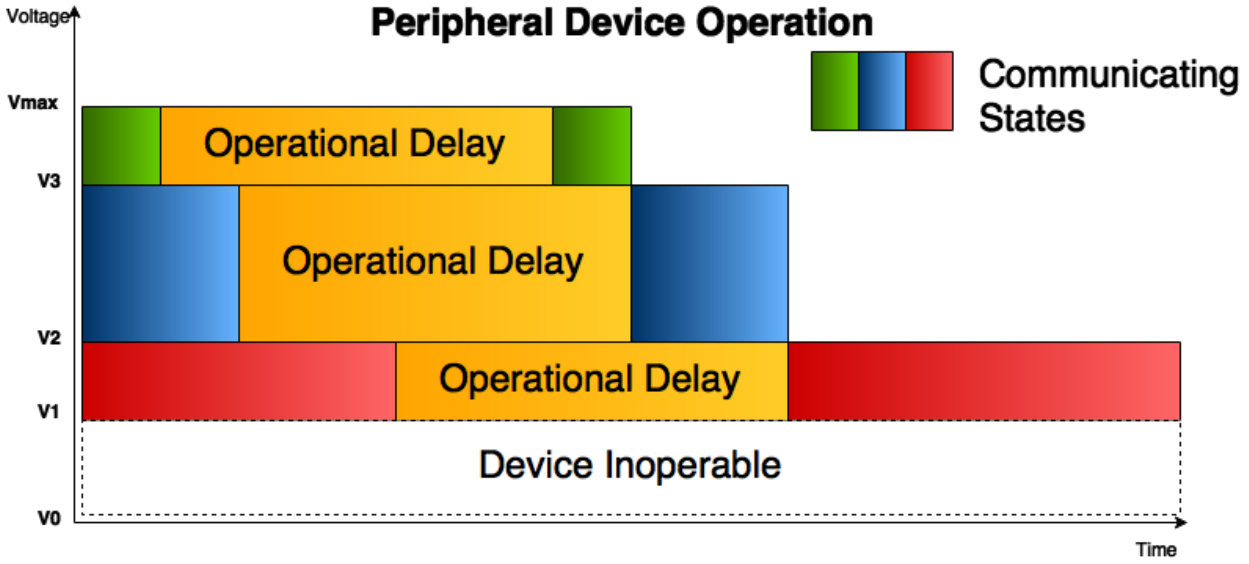


Figure 3: Voltage Dependent / Independent Device States

Devices may exhibit voltage independent behavior due the presence of an onboard internal regulator such as in the case of various EEPROM and flash peripherals. Voltage-independence may also exist due to physical characteristics of a peripheral sensor, such as the time required to accumulate a measurement (photons, gas, electrons, etc.). In any case, this is an opportune time to exploit voltage-independence and decrease power consumption throughout the duration of the operation.

Devices such as those shown in Figure 3 exhibit the energy and delay characteristics shown in Figure 4. Increasing the signaling/supply voltage to the device initially results in a sharp decrease in communication delay as described in the previous section. Diminishing marginal returns occur as the peripheral is bounded by internal limitations. However, increasing the signaling voltage also results in an unnecessarily energy consumption during the voltage-independent state. The ultimate result is that an unnecessarily high energy penalty is paid for marginal increases in performance.

The effect of increased voltage throughout the operation of the peripheral is shown in Figure 4. The overall energy consumption increases exponentially while decreases in response time are marginal. This occurs because only the voltage-dependent states are eligible for the performance increase while the power usage penalty is paid throughout all states.

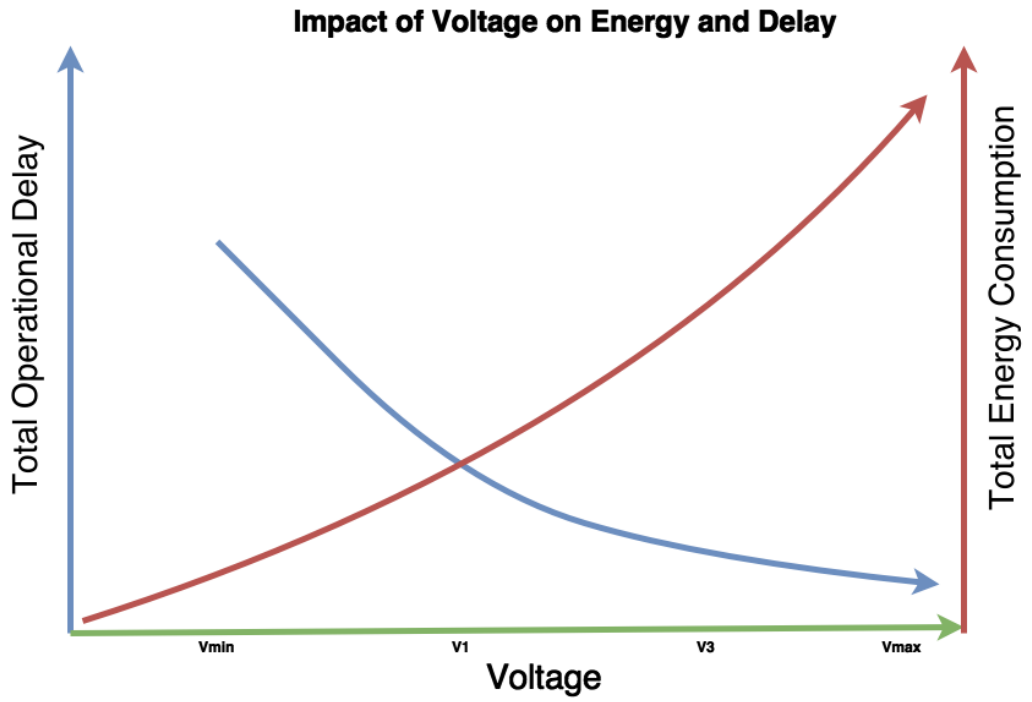


Figure 4: Impact of Voltage on Energy and Delay

Intra-Operation Dynamic Voltage Scaling

Performance of a peripheral device is maximized by operation at its maximum voltage and frequency during voltage-dependent states. Energy consumption of a device is minimized by operation at its lowest possible voltage during voltage-independent states. Such a system would result in transforming the operation shown in Figure 3 into the same operation shown in Figure 5.

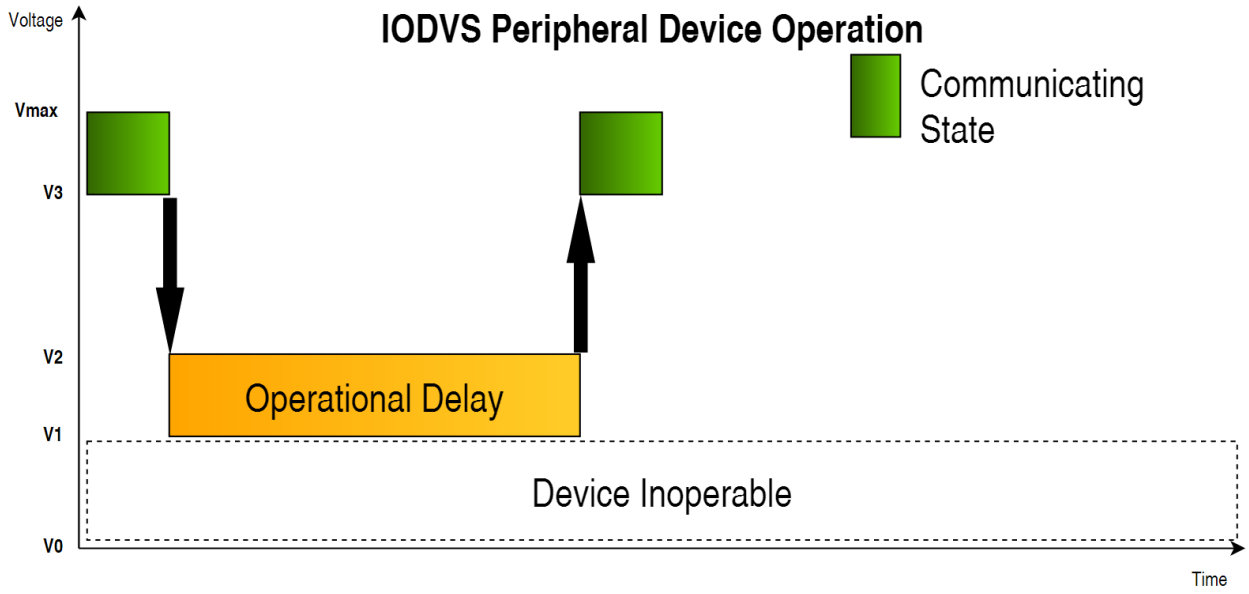


Figure 5: IODVS Peripheral Device Operation

The supply voltage of the device is manipulated so as to minimize the duration of voltage dependent states and to minimize the power draw of voltage independent states. These transitions occur as peripheral

devices are carrying out operations such a memory accesses or environmental measurements and thus the voltage scaling occurs intra-operation.

By merely implementing IODVS and measuring the effect, it is shown that further timing optimizations can be made to the operation of peripheral devices. By observing current consumption during the operation of a peripheral device, it can be more accurately estimated as to when the device has completed the operation. This information is used to build both timing and energy consumption heuristics in order to determine completion and resume operation earlier than specified by the manufacturer.

Achieving the optimal power profile for one device may result in affecting the operation of other devices that are powered from the same domain. To combat this, a supervisor is implemented and tested in order to prevent conflicts from varying device voltage requirements. The supervisor uses multiple methods of varying complexity to determine eligibility for voltage transition.

The remainder of this research investigates the benefits of voltage scaling at such a fine granularity. Energy and delay requirements are considered system-wide in order to exploit the potential savings offered by IODVS. The system is defined as a combination of power supplies, a governing micro controller and a variety of peripheral devices. Specific consideration is given to maximizing the overall efficiency with which peripheral device operations are performed. This work focuses on extracting those efficiency gains by performing the following investigations:

1. System definition:
    a. The application microcontroller (MCU). With attention to dynamic voltage and frequency scaling (DVFS) and dynamic power management (DPM) capabilities.
    b. The characteristics of peripherals attached to the MCU. Fine-grained dynamic loading characteristics are of particular interest.

2. Intra-Operation Dynamic Voltage Scaling (IODVS)
    a. Fine-grained modulation of peripheral power supplies enables significant energy savings with no-effect on peripheral performance.

3. Activity Completion Recognition (ACR)
    a. Complete set of peripheral operations performed while metering current consumption and operation duration across multiple devices, voltages and temperatures.
    b. Time and current based adaptive heuristic for early-completion estimation.
    c. Integrated energy based heuristics for early completion estimation.

4. Supervised IODVS
    a. Reduce IODVS domain interference by identifying interfering voltage ranges.
    b. 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.

IODVS is shown to reduce energy consumption in many common peripheral devices by 10 – 40% depending on the ratio of voltage-dependent to voltage independent states. In-system metering of peripheral devices is allows the MCU to detect operation completions and thus decrease the duration of voltage-independent states. Finally, the procedure is generalized for easy implementation in most embedded systems through the development of a supervisor which arbitrates the voltage demands of peripherals that share a voltage domain.

CHAPTER 2: SYSTEM DEFINITION AND RELATED WORK

Energy management is performed by investigating the efficiency and capability of the power supplies as well as the loading characteristics of the energy consumers. This information enables traditional power management algorithms such as Dynamic Voltage and Frequency Scaling (DVFS), or Dynamic Power Management (DPM) to make real-time adjustments in pursuit of efficiency. Comprehensive system information enables IODVS to operate at extremely fine granularity.

Power Supplies

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. On a more practical level, the higher supply voltage also 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.

Linear / Low-Dropout Regulator (LDO)

The least complex circuit for step-down applications is the linear regulator shown in Figure 6. 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 $V_{dropout}$. Modern versions of the circuit have focused on decreasing this margin and are known as LDOs (Low Dropout Regulators).

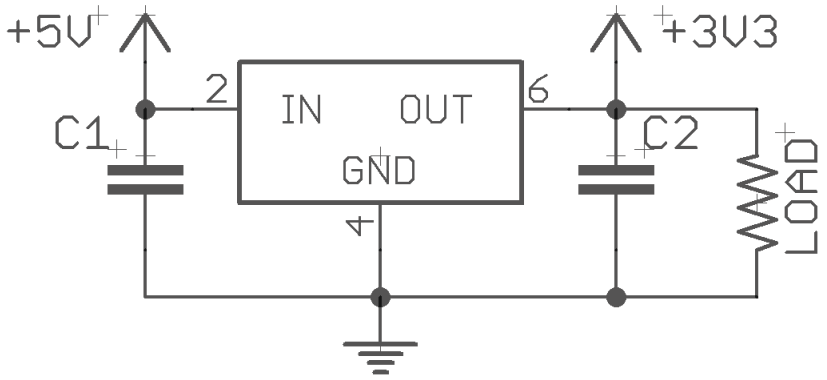


Figure 6: A Linear Regulator / LDO Circuit

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 converted and quiescent loads: $P_{Reg} = (V_{in} - V_{out})I_{out} + V_{in}I_Q$. The quiescent current is usually so low as to be ignored and thus the efficiency of the converter can be approximated to $\eta = \frac{V_{out}}{V_{in}}$. 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 provides more efficient translation of large voltage differentials than the LDO. Figure 7 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 switched mode power supply 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.

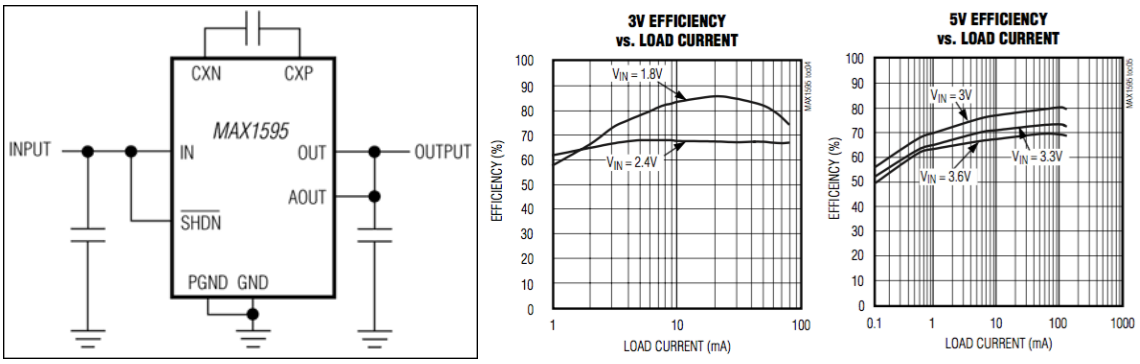


Figure 7: A Typical Charge Pump Circuit and Efficiencies [2]

Switched Mode Power Supply (SMPS)

For systems requiring DC-DC conversion (contrasted with voltage translation by increased current capability), the SMPS is the most common application. For the majority of embedded systems a circuit is required to step down voltage from a high level to a lower level via the buck configuration as shown in Figure 8:

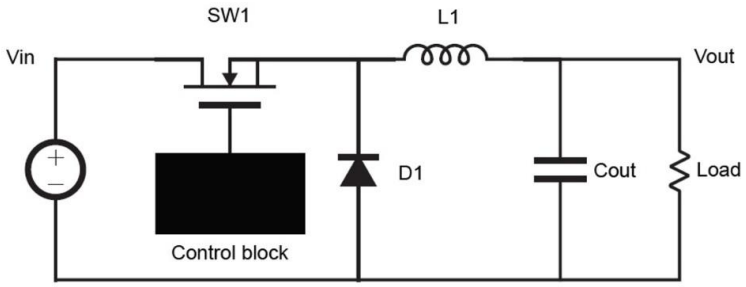


Figure 8: A Simple SMPS in Non-Synchronous Buck Configuration [3]

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 generally 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 I^2R 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.

Switched mode power supplies can be configured in the step-down buck configuration, the step-up boost configuration, or a combination buck-boost configuration which operates independent of input voltage. In any case, the increased current sourcing capabilities come at the cost of having the most PCB area of the options considered. Also, in all cases, the SMPS produces a ripple voltage at the output which designers strive to minimize through filter circuitry thus increasing total bill of materials cost. Many sensing

devices are sensitive to disturbances caused by ripple and thus it is a variable that must be minimized or eliminated through the application of an additional LDO.

Energy Management Techniques

Embedded energy management research to date is split into two distinct fields: Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS). DPM policies tend to focus on strict power-state relationships [4], while DVS policies tend to incorporate a linear power-performance relationship [5].

In fact, DVFS is so useful that hardware is designed specifically to take advantage of it [6]. Most DPM implementations focus on optimal scheduling techniques such that peripherals emerge from low-power states just in time for access by tasks requiring their functionality. Generally, the approaches to date can be categorized as a combination of either online [7] or offline [8] and deterministic [9] or probabilistic [10]. Dual-output circuits have been developed primarily targeting systems with a SoC [11]. The same circuitry could be reused to implement IODVS.

With respect to peripheral power management, peripherals can be operated under the same linear, DVFS based constraints [14] as wells as step-wise DPM based [15] 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) [10]. Similar resource availability problems are encountered with IODVS and similar heuristics are applied to address them.

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 minimize the increase in latency.

The break-even time is the figure of merit for DPM as it pertains to energy savings [12]. It is defined as the duration that a device must remain asleep in order to offset the energy spent waking the device throughout which duration the device will be inaccessible. 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 disabling it.

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 [8]. Similar data can be realized online by profiling tasks and determining which paths lead to a peripheral access [13]. Both methods enhance the accuracy of predictions regarding the optimal peripheral wakeup time. All methods evaluate the cost/benefit of peripheral deactivation with respect to the energy savings gleaned versus the time spent reactivating the device when next needed.

Dynamic Voltage (and Frequency) Scaling

Microcontrollers use power at the rate $P_{MCU} = P_{dynamic} + P_{static}$. Static power dissipation P_{static} is due mostly to leakage currents throughout the MOSFETs of the MCU. Dynamic power dissipation $P_{dynamic} = fCV_{dd}{}^2$ where f is the switching frequency of the circuit and C is the MOSFET gate capacitance of active circuits. The switching voltage V_{dd} is ripe for optimization because power consumption is proportional to the square.

In addition to the substantial power savings afforded by decreasing V_{dd}, 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. Optimal systems are operated at a clock frequency exactly sufficient to complete all tasks by their deadlines. Likewise, a sufficient supply voltage must be applied in order to achieve that clock frequency.

Wireless Sensor Networks

Minimalistic embedded systems such as wireless sensor networks (WSNs) are tasked with sensing their environment and communicating their readings to a more capable host for processing. Their power requirements are low because processing is typically offloaded to more capable nodes with more reliable power supplies. This is usually accomplished via a mesh network that grows as new nodes find and establish local communication with one another [16].

They are often powered from renewable sources or long term batteries, in some cases lasting over 10 years [17]. The responsibilities of a node on the network are minimized so as to achieve such a long lifespan. Dynamic voltage scaling techniques have been employed to decrease the energy consumption of these devices [18]. Due to the step-wise nature of their task sets, WSNs respond better to DPM schemes as the energy management technique, with DVFS employed during the active period. These systems are an excellent example of where IODVS would be ideal because of their typically short duty cycles.

Component Aware Dynamic Voltage Scaling

IODVS is most similar to the Component Aware DVS technique [16] [17] developed for use in the nodes of a wireless sensor network [19]. An adjustable regulator is operated by the MCU in an embedded system such that it is operating at its minimum voltage requirements. 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 and therefore intra-task granularity.

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 standby mode that allows the system to drastically decrease the static power consumption of peripheral devices. Thus, a significant amount of research has gone into determining the optimal breakeven time of embedded peripherals. DPM techniques inherently interfere with the operation of the device and impose a lag in response time.

IODVS, while maintaining compatibility with DPM, instead exploits the acceptable operating voltages of the device and does so with no effect on response time. IODVS is primarily beneficial to devices with the same responsibilities and characteristics as those of a node on a WSN. Therefore the peripherals under consideration are likewise responsible for sensing, storing and communicating. Each type of peripheral has both voltage-dependent and voltage-independent states due to the characteristics of the device.

Peripheral storage devices such as EEPROM and flash tend to incorporate onboard voltage regulators so as to ensure a reliable supply during read and write operations. Embedded sensors often incur a voltage-independent state during a sensing operation because the sensor requires the medium to accumulate for a period of time before an accurate measurement can be made. Communicating peripherals such as wireless transmitters also incorporate voltage regulators and buffers. This is necessary because the output voltage of a wireless transmitter must be maintained within strict parameters and the output message must be transmitted within strict temporal limits.

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 Figure 9. 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. It is demonstrated that energy can be saved by lowering the peripheral domain voltage during voltage independent states 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.

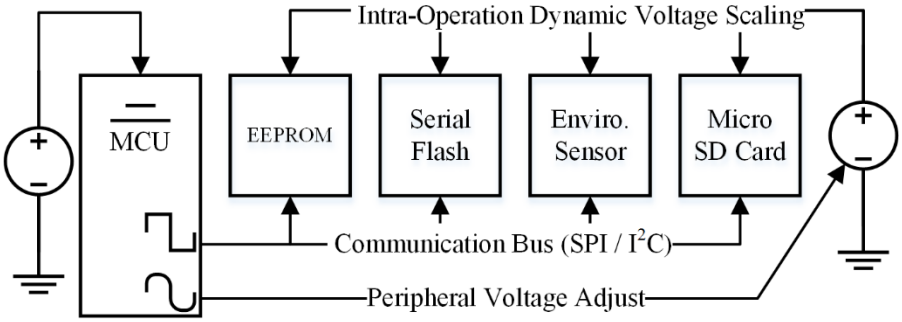


Figure 9: An IODVS Enabled System

For example, EEPROM is a typical peripheral device that is used to provide non-volatile data storage. The devices 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 Figure 10. 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 coordinated voltages, thereby maximizing data transfer, minimizing energy delay product (EDP) and eliminating the need for voltage level translation.

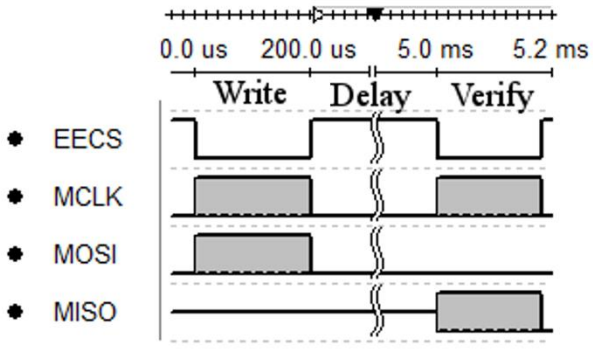


Figure 10: A SPI EEPROM Write / Verify Cycle (Not to Scale)

The most distinct benefit of IODVS can be realized during the longest portion of the typical transaction described in Figure 10: the delay. IODVS decreases the supply voltage to the chip during this voltage-independent period and it is demonstrated that the total energy cost of the transaction is significantly decreased.

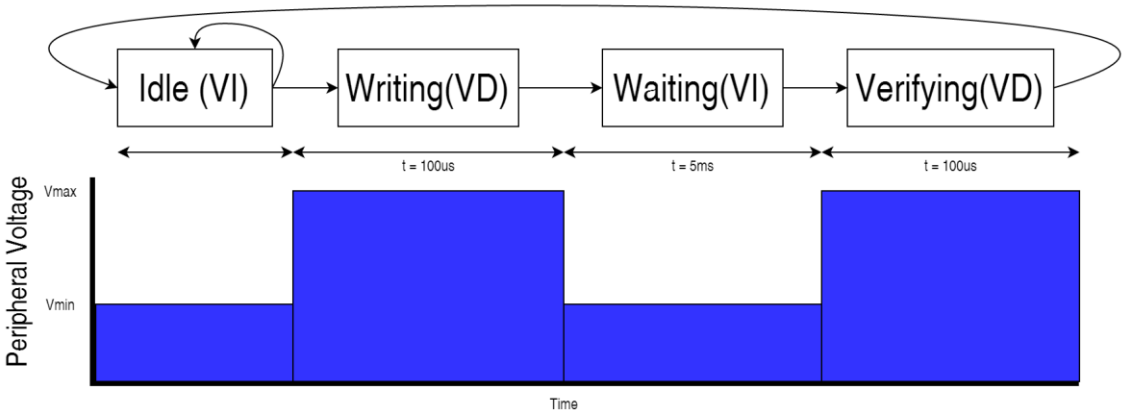


Figure 11: EEPROM Write Operation State Diagram and Corresponding Voltage /Time Relation

IODVS is implemented by creating a state transition diagram for each operation that a device may perform and noting the voltage-dependent (VD) and voltage-independent (VI) states. Figure 11 shows the example state transition table for write operation to EEPROM. Vmin is specified by the datasheet as the voltage at which the device will cease to operate predictably. Vmax is voltage capable of providing maximum performance throughout the state. It is bounded by the lower of either where the device ceases to increase in performance or by the voltage at which the MCU is unable to communicate with the peripheral.

In order to write to memory, the device transitions from the VI idle state, into the VD writing state where the MCU is sending data to the device. The transaction is voltage dependent because the communications performance scales with voltage. From the writing state the device transitions into the VI waiting state. The waiting state is specified by the datasheet to be 5ms regardless of applied voltage and is therefore, by definition, voltage-independent. The MCU then reads back the data to ensure integrity it was committed properly. This ‘verifying’ state is voltage-dependent because the MCU and peripheral are communicating throughout.

The operational state diagrams can be used to create a Peripheral Power Profile (PPP) for each device. This PPP forms a lookup table of state-voltage pairs. Combining the operational state transition table of Figure 11 with the Vmin and Vmax determined by the datasheet and MCU (summarized under MCP25AA512 in Table 3) the PPP for the device is formed in Table 1:

Table 1: PPP as Derived from State Diagram

State	Voltage
Idle	1.8v
Writing	3.3v
Waiting	1.8v
Verifying	3.3v

It can be beneficial to estimate the energy savings of implementing IODVS at design time against the cost in both design effort and bill of materials. This is a remarkably difficult estimate to create as the current consumption dynamics of the Results section illustrates. A reasonable estimate can be made by comparing the current consumption of the device at Vmin against the current consumption at Vmax and accumulating power consumption throughout the duration that the device spends in each state as described in (5).

$$E = \sum_{s=0}^{s-1} V_s I_s T_s \quad (1)$$

Returning to the example of a typical EEPROM the data in Table 2 are found within the device specification or in some cases either extrapolated or interpolated:

Table 2: Estimated State Voltage Current and Duration Pairs

State	Current @3.3V	Duration @3.3V	Current @1.8V	Duration @1.8V(Est.)
Idle	Not Provided	Steady State	Not Provided	Steady State
Writing	6.0mA	~200us	4.5mA	~1000us
Waiting	Not Provided	5ms	Not Provided	5ms
Verifying	7.5mA	~200us	3.0mA	~1000us
*Note that current and duration are average estimates from the device datasheet				

Because idle current consumption was not provided for varying voltages, the idle state is removed from the estimate. Thus, the estimate will reflect a device that is operating 100% of the time (never returning to idle). Likewise, the mandatory wait period current consumption was specified as an average occurring throughout the writing state. For the purposes of estimation, the wait state is combined with the writing state. Noting the latency increase due to the decrease in signaling voltage, these estimates result in:

$$E_{3.3V} = 3.3v * 6ma * .2ms + 3.3v * 7.5ma * 5.2ms = 132.66\text{uJ} \quad (2)$$

$$E_{1.8V} = 1.8v * 4.5ma * 1ms + 1.8v * 3.0ma * 6ms = 40.5\text{uJ} \quad (3)$$

The energy result of (3) is the lower bound on the energy that a peripheral operation can consume without IODVS while the 5.4ms duration of (2) is the lower bound on operation latency. Without IODVS, the latency decrease of 1.6ms is paid for through the energy increase of 92.16uJ. Because the manufacturer specified only an average current throughout the write operation and knowing that IODVS will lower the voltage to Vmin throughout the course of the delay portion of the write operation, we can estimate between the two bounds as:

$$E_{IODVS} = 3.3v * 6ma * .2ms + 1.8v * 3.0ma * 5.0ms + 3.3v * 7.5ma * .2ms = 35.91\text{uJ} \quad (4)$$

When compared against constant latency, the change in energy consumption from (2) to (4) is decreased by 73%. For cases where latency is irrelevant, comparing (3) to (4) yields energy savings of 11.3% due to the decrease in time spent in voltage-dependent states. The resulting estimation of (4) is likely to contain inaccuracies due to both the current and voltage dynamics of devices on the domain. Specifically, this estimation assumes instantaneous voltage changes between states which may or may not be accurate depending on the capacitance of the domain and the current consumption of the domain at the switching time. System level losses such as pull-up resistors and trace impedances will also affect the actual results.

The IODVS technique is applicable to any peripheral that has a voltage/performance dependence and particularly applicable to those with a wait-state. The investigation considered the peripherals listed in Table 3 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.

Table 3: Typical External Peripherals

	Honeywell HIH-6130 I^2C 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). 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. Idle state energy decreases of up to 66% and intra-operational state energy decreases of up to 40% were observed.

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 [20] 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 were 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.

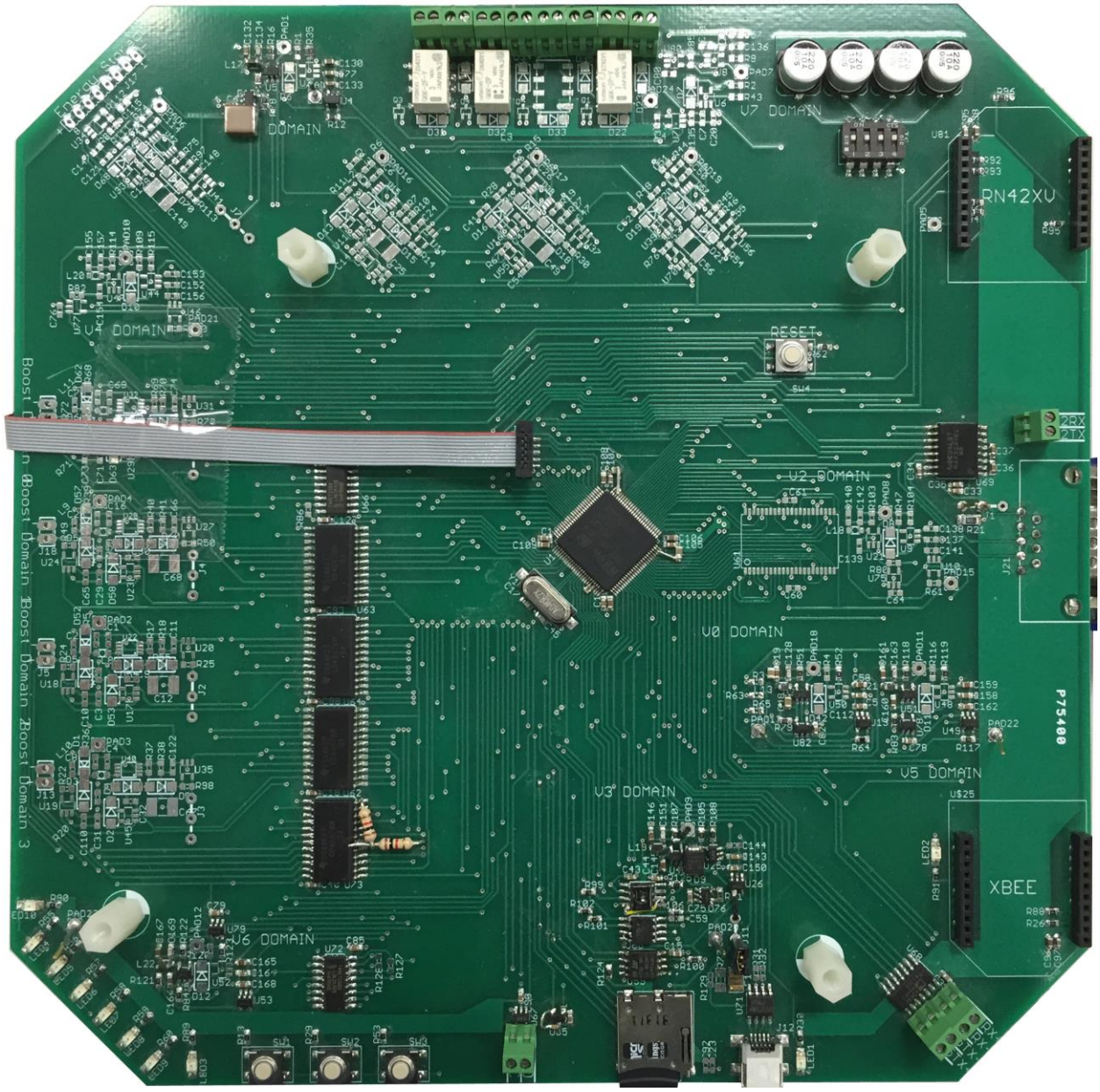


Figure 12: Peripheral Generation Measurement and Allocation (PEGMA) Circuit Board

Methods and Materials

The TPS62240 [21] 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 [22] on both the input to the SMPS and the output to the domain.

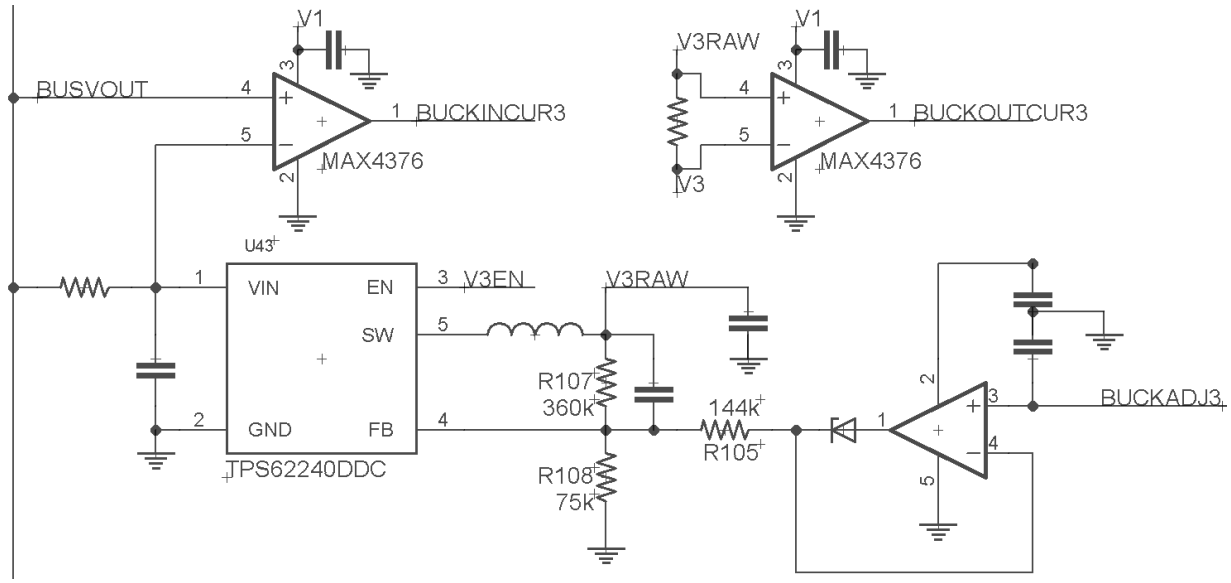


Figure 13: Peripheral Domain SMPS, Control and Current Sense Circuitry

As shown in Figure 13 and expanded upon in Appendix A, 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 the 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 (5). The results were calculated offline and digitally integrated via (6) and (7), where S is the state of the device, and T_s is the sampling period.

$$P = VI = \frac{E}{t} \tag{5}$$

$$E_s = \sum_{n=0}^{N-1} V_n I_n T_s \tag{6}$$

$$E_{total} = \sum_{S_0}^{S_{n-1}} E_s \tag{7}$$

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.

Likewise, IODVS is applicable to an exploitable sequence of active operations, resulting in decreased energy consumption without the performance impacts of DVFS. Energy consumption can be separated into two intervals as shown in (8).

$$E_{total} = E_{idle} + E_{active} \tag{8}$$

$$E_{total} = P_{idle} * T_{idle} + P_{active} * T_{active} \tag{9}$$

A duty cycle of 0% will be dominated by T_{idle} and energy consumption will converge on that of the idle state. On the other hand, a duty cycle of 100% will be dominated by T_{active}. 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 [23]. 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. Figure 15 provides a comparison of both the standard PPP and the 1.8VIW profiles enabled by IODVS.

The state transition diagram of Figure 14 is known a-priori and is followed throughout the tests illustrated in Figure 15. 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 Figure 15, 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%.

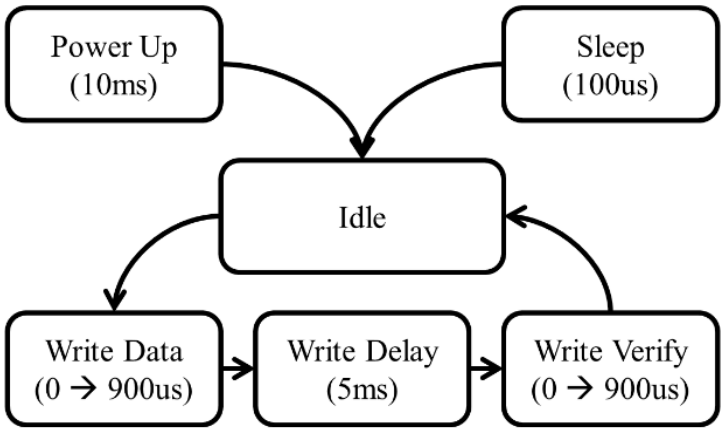


Figure 14: EEPROM Write State Transition Diagram

Table 4: MCP25AA512 Peripheral Power Profile

State	Voltage (Control)	Voltage (IODVS)	Duration
Idle	3.3v	1.8v	Steady State
Writing	3.3v	3.3v	~500us
Waiting	3.3v	1.8v	5ms
Verifying	3.3v	3.3v	~1ms

Table 5: MCP25AA512 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%**

Table 6: MCP25AA512 Energy Consumption and 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%

The EEPROM does implement an optional sleep state that incurs a 100us penalty from which to wake. Systems that optimized for response time (such as in the case NVRAM) may not be capable of waiting 100us and therefore IODVS is attractive for the 66.7% power consumption reduction without incurring the wake penalty. For systems that are capable of withstanding the wake penalty, the PPP would be modified to include a 1.8v sleep state which would further reduce energy consumption of the device.

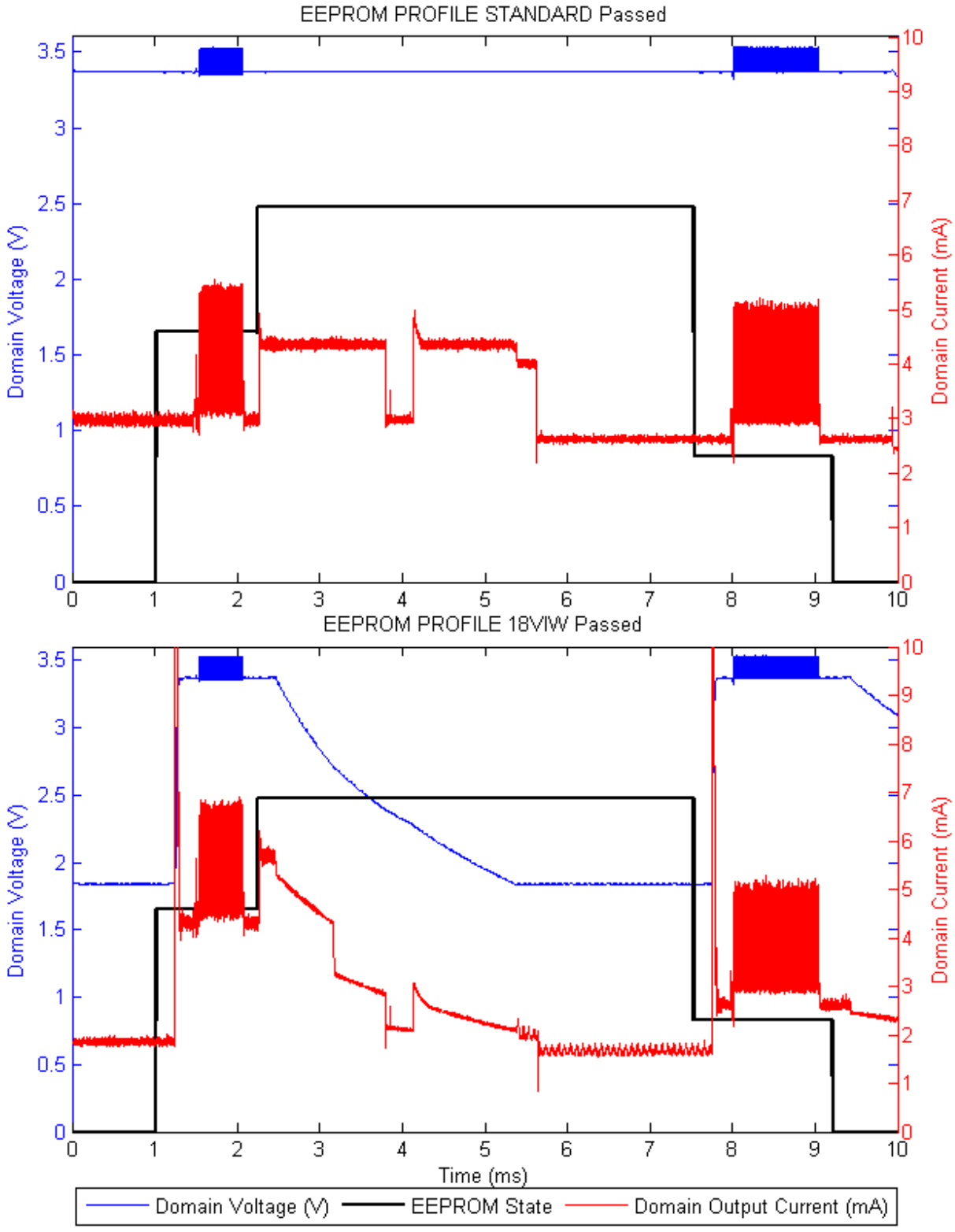


Figure 15: EEPROM IODVS Test

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 [24] 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 Figure 16 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 Figure 17. 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.

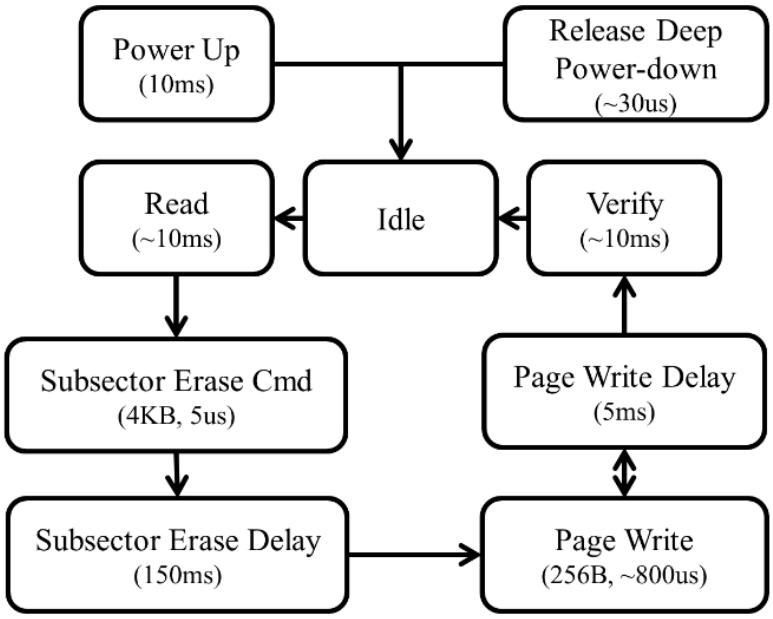


Figure 16: Serial Flash Write State Transition Diagram

Table 7: M25PX16 Peripheral Power Profile

State	**Voltage (Control)**	**Voltage (IODVS)**	**Duration**
Idle	3.3v	2.3v	Steady State
Reading	3.3v	3.3v	~10ms
Erase (Command)	3.3v	3.3v	~10us
Waiting	3.3v	2.3v	~150ms Erase ~5ms per Page
Writing	3.3v	3.3v	~1ms per Page
Verifying	3.3v	3.3v	~10ms

Table 8: M25PX16 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

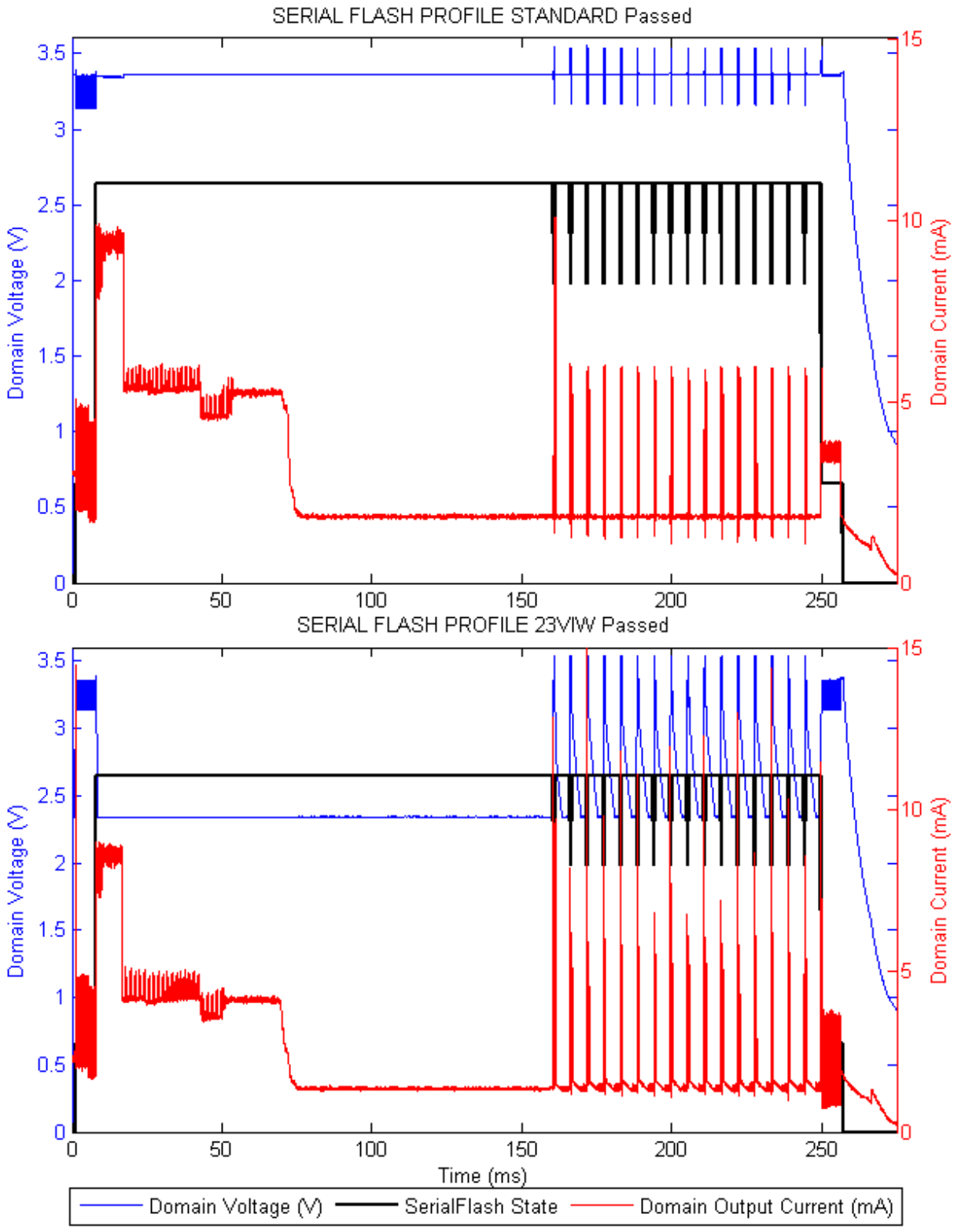
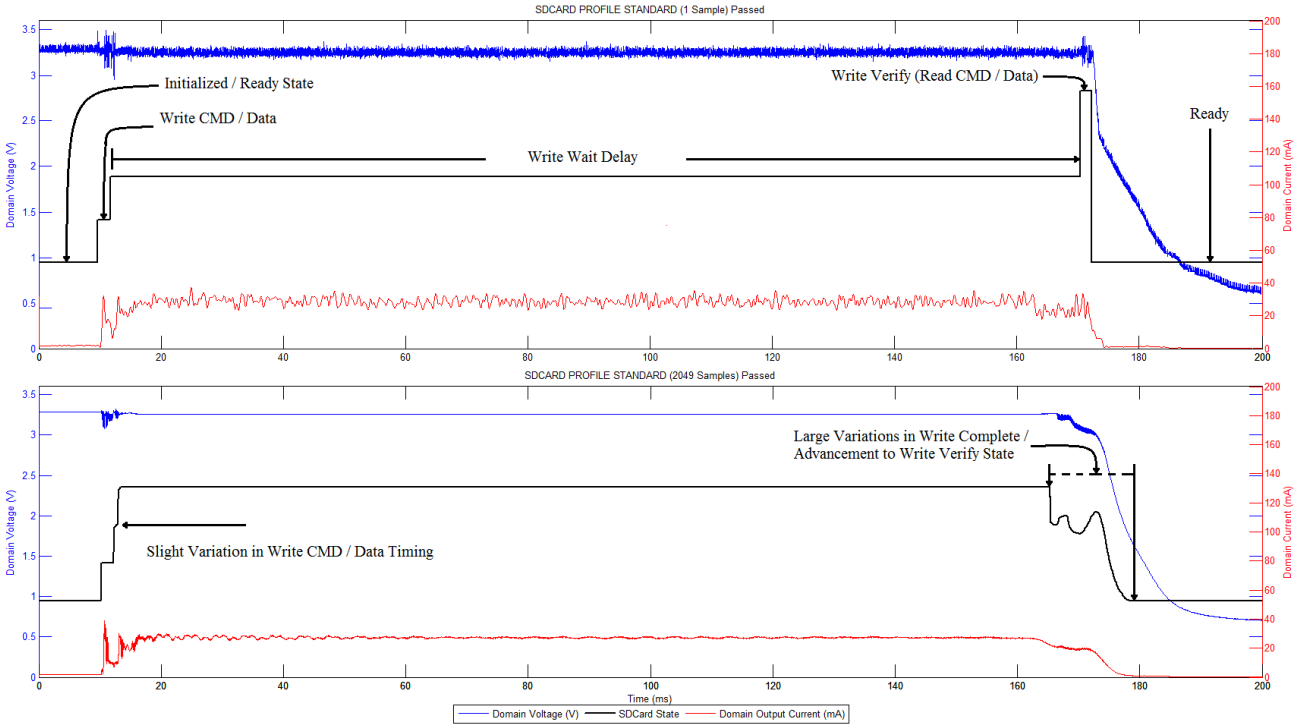


Figure 17: Serial Flash IODVS Test



Micro-SD Memory Card

Figure 18: Typical Micro-SD Memory Card Test

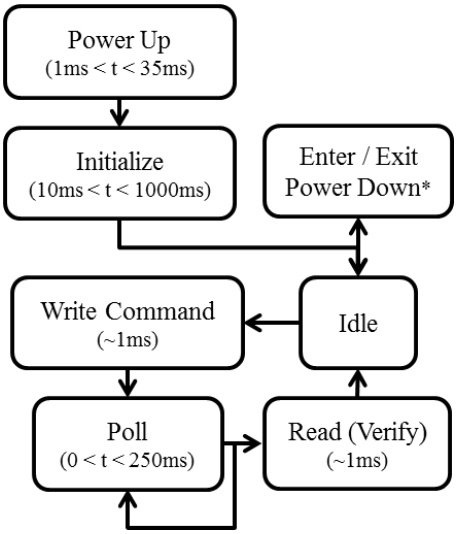


Figure 19: Micro-SD Memory Card Write State Transition Diagram

Micro-SD Cards follow a standard outlined by the SD Association [25]. The standard is a minimum set of electrical and communication specifications that must be met and some vendors exceed those specifications [26] [27] [28]. 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. IODVS assumes matched voltages during MCU to device communication periods. Polling during write operations can be avoided by predicting the write completion time.

Figure 18 shows how timing variations affected the 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 addressed in the following chapter.

An SD Card must be initialized after power up as shown in Figure 19. 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.

Table 9: Generic Micro-SD Memory Card Peripheral Power Profile

State	Voltage (Control)	Duration (IODVS)	Duration
Idle	3.3v	2.7v	Steady State
Write Cmd (Polled)	3.3v	3.3v	~10ms
Write Data	3.3v	3.3v	~10us
Waiting	3.3v	2.7v	~(10 – 150ms)
Write Complete?	3.3v	3.3v	~10us
Verifying	3.3v	3.3v	~10ms

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 Figure 20, 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, it was found 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.

Table 10: 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 in Figure 21 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.

Table 11: 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%**

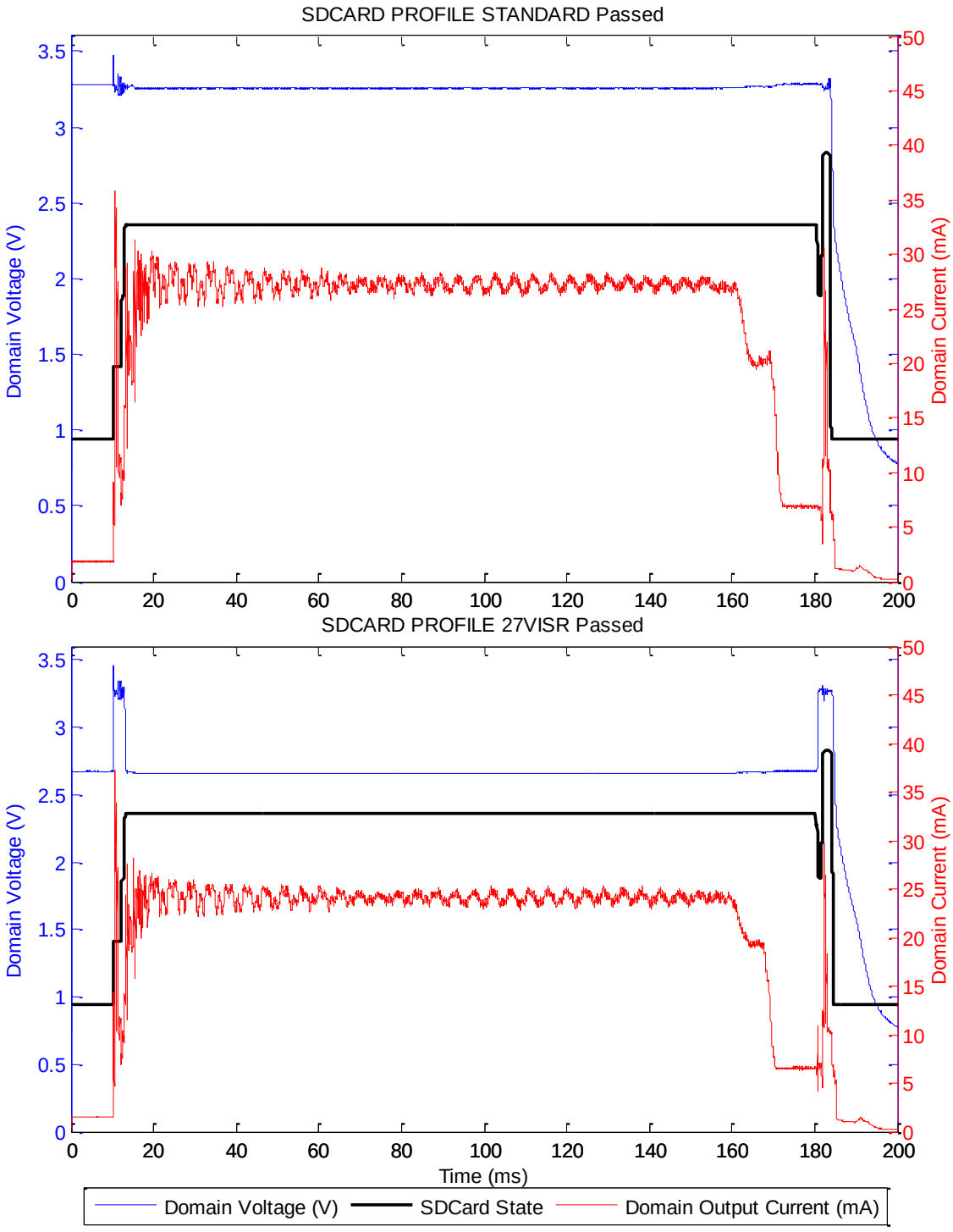


Figure 20: Sandisk Micro-SD Card IODVS Test

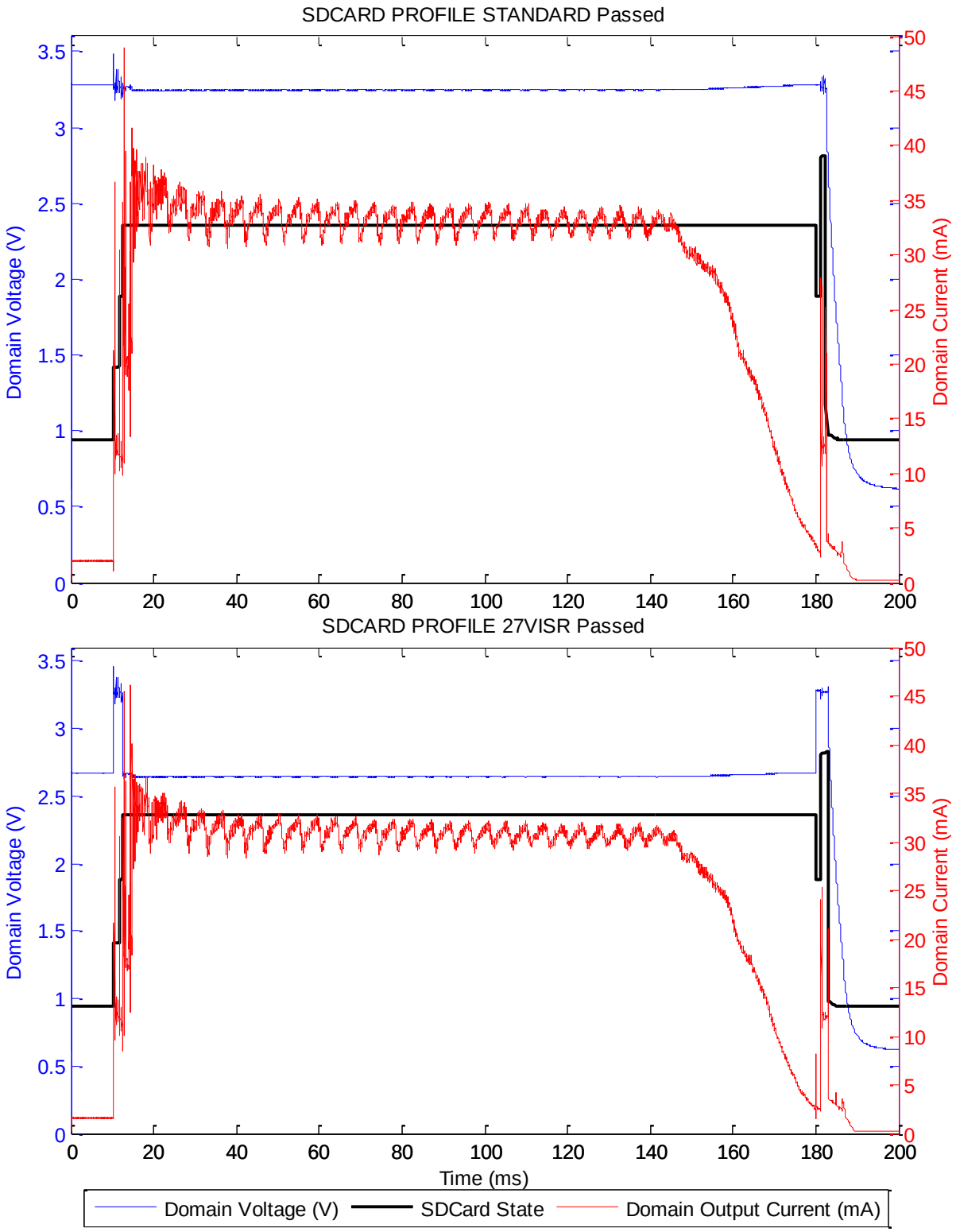


Figure 21: Lexar Micro-SD Card IODVS Test

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 Figure 22 indicates that writes begin completing at approximately the 35ms mark.

Table 12: 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%.

Table 13: 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%**

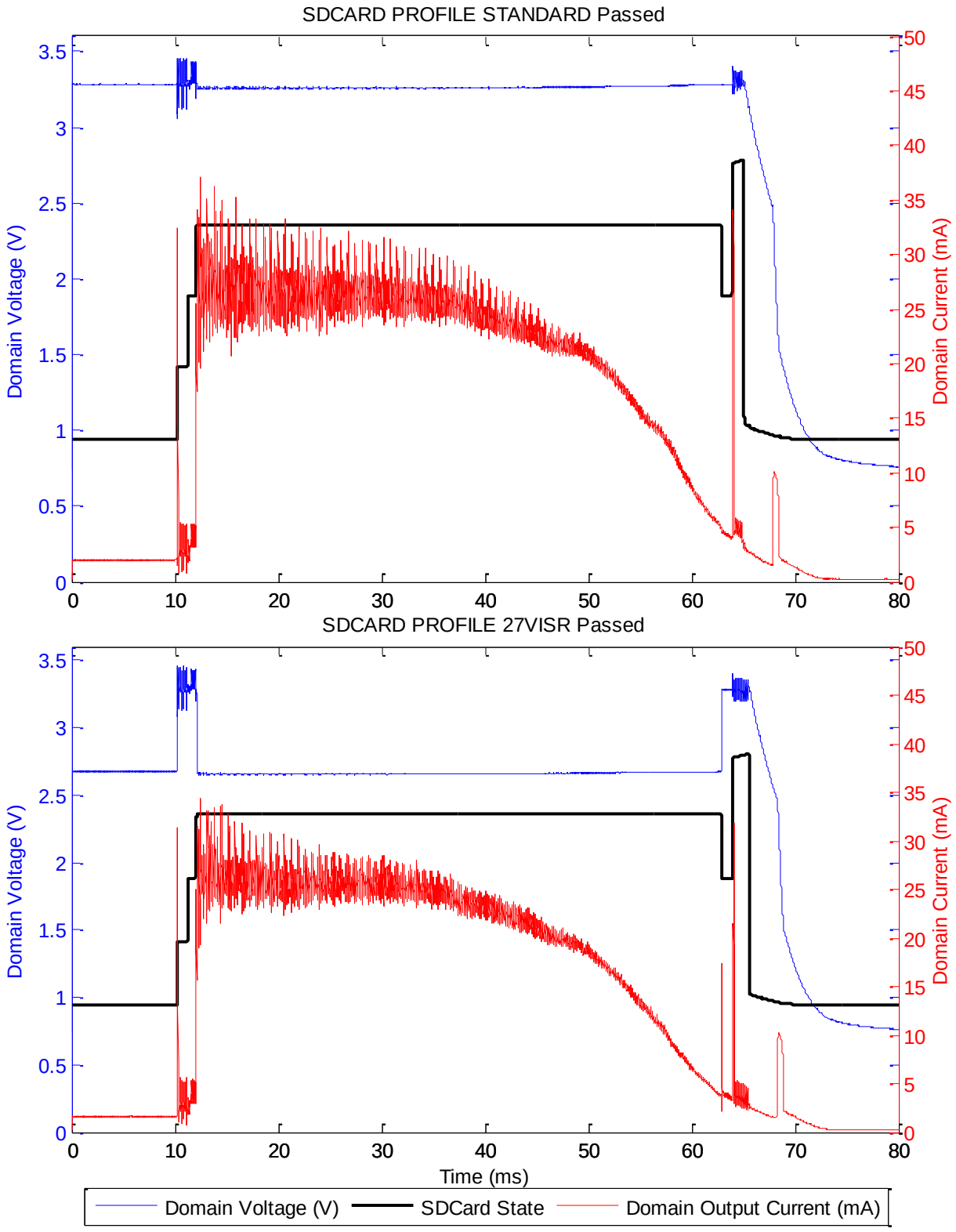


Figure 22: SwissBit Micro-SD Card IODVS Test

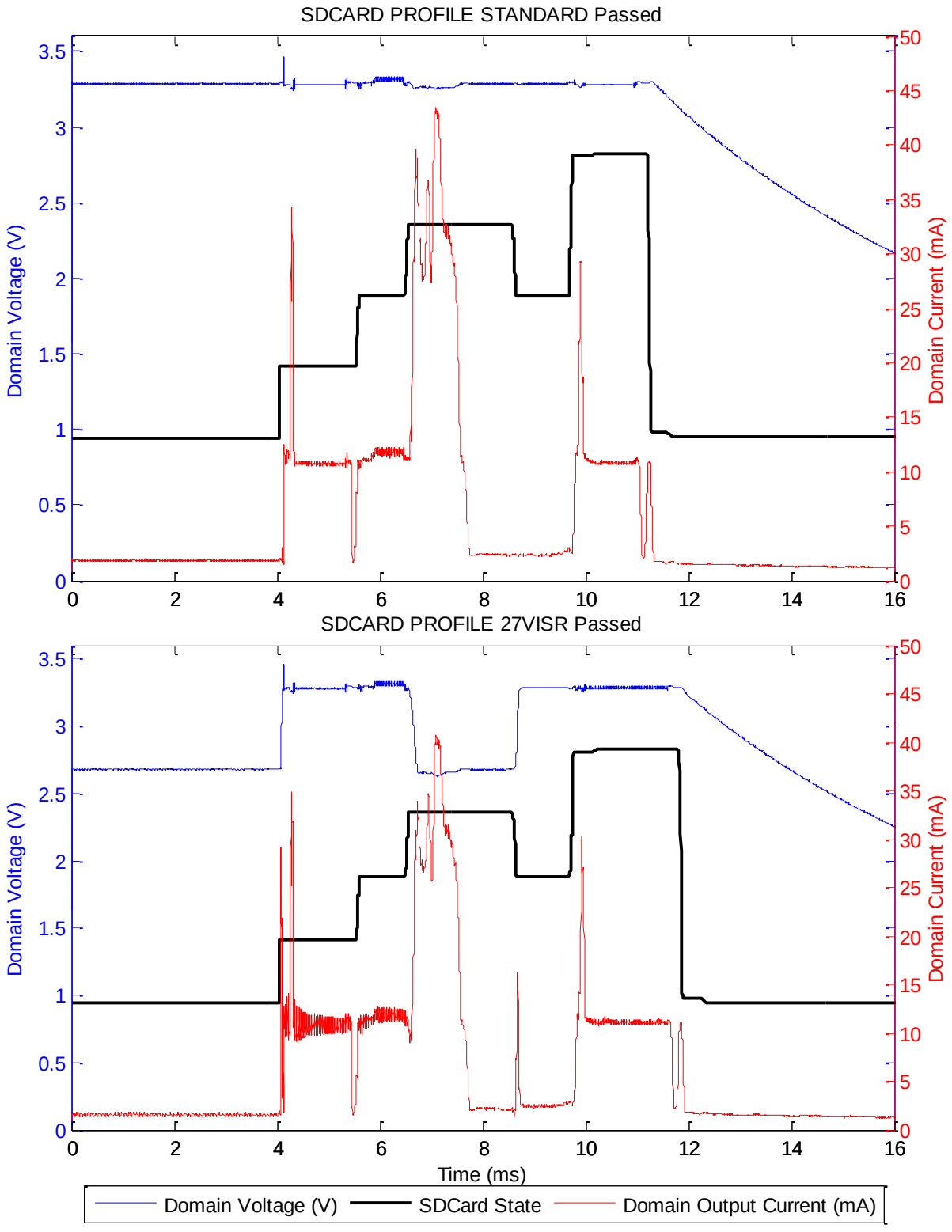


Figure 23: Kingston Micro-SD Card IODVS Test

Honeywell HIH6130 Temperature / Humidity Sensor

The MCU communicates with the temperature and humidity sensor [29] via I^2C. 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 I^2C 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 I^2C 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 Figure 24 and the MCU issues a “Measure” command to the sensor. The peripheral takes up to 4ms to wake from sleep [30] and then transitions to the temperature measurement and humidity measurement states in sequence. There is a noticeable drop in current in Figure 25 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 duration of the reading state was unaffected by the optimization because voltages are equal.

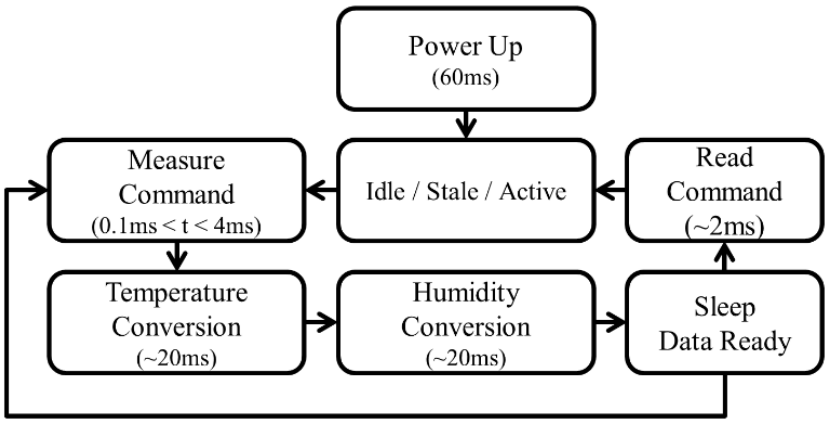


Figure 24: HIH-6130 State Transition Diagram

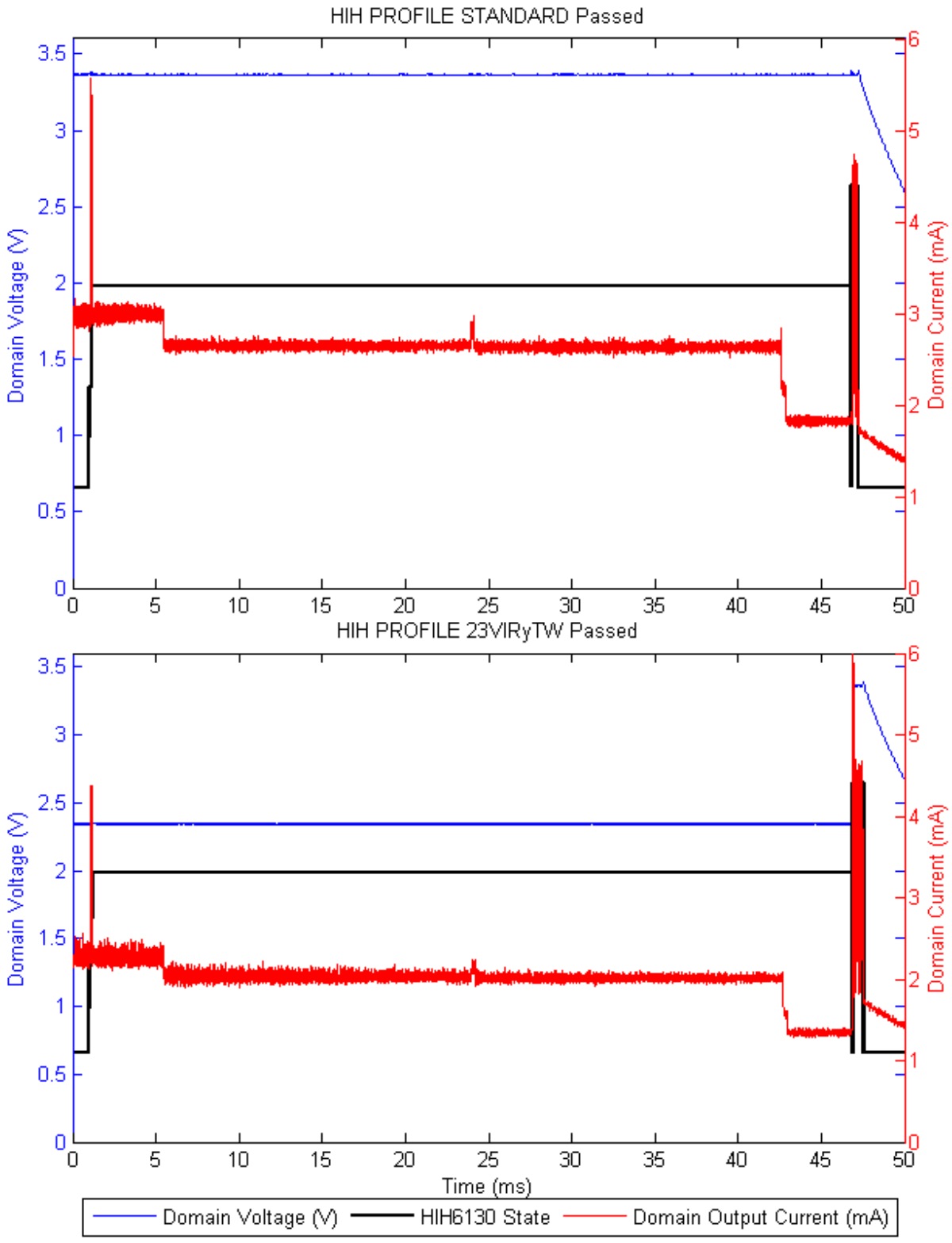


Figure 25: HIH-6130 Temperature / Humidity Sensor IODVS Test

Table 14: HIH-6130 Peripheral Power Profile

State	Voltage (Control)	Duration (IODVS)	Duration
Idle	3.3v	2.5v	Steady State
Measure Cmd	3.3v	3.3v	~100us
Waiting	3.3v	2.5v	~45ms
Reading	3.3v	3.3v	~1ms

Table 15: 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%**

Conclusion

IODVS has been shown to decrease energy consumption on a typical group of external peripherals by 10-40% with no decreases in either performance or accuracy. The efficacy of the technique tends to increase with low-duty cycles which is typical of external non-volatile memory. The CPU overhead of performing IODVS is negligible through the use of pre-defined peripheral power profiles.

Minimal additional circuitry is required to implement IODVS. In many cases the decrease in power budget or increase in performance may offset the additional cost. These experiments made use of a DAC because of the flexibility it offered in voltage modulation for a wide variety of devices. Simpler implementations would benefit from switching SMPS feedback resistors into and out of the circuit.

The technique would be most effective if it were used in a system with minimal domain capacitance. This would allow for faster changes in domain voltage which would reduce the response time of the SMPS and reduce the inrush current when charging a domain. Ideal domain capacitance is a balancing act between IODVS dynamics and against the peripheral load dynamics.

All of the devices tested provide a mechanism for testing operation-complete. We used the timing specifications contained within the device datasheet for all devices except the Micro-SD Cards (where polling was mandatory). Based on the current profiles of the devices, it can be inferred that most operations completed earlier than the datasheets specified. It would be worthwhile to pursue further research combining operation-completion prediction heuristics with IODVS to further minimize energy consumption.

Manipulating the voltage across a domain of devices is bound to impact some devices more than others. For instance, if the domain voltage drops below 2.7V, some SD Cards may revert from an initialized state to the idle state. Therefore, before adjusting a particular device on the domain, IODVS should determine if that would cause an overall benefit or detriment to devices on the domain. This could be determined by majority vote of devices on the domain and DPM-inspired analysis could be used to aid in decision making. If indeed the voltage is manipulated out of bounds for a particular device, the driver for each device on the domain needs to be notified of the voltage change so that re-initialization can take place if necessary.

CHAPTER 4: ACTIVITY COMPLETION RECOGNITION (ACR)

Introduction

Intra-Operation Dynamic Voltage Scaling (IODVS) has been shown to significantly reduce the energy consumption of embedded peripherals during their voltage-independent states. These states typically occur during mandatory delay periods as the device completes a specified operation. Activity Completion Recognition (ACR) seeks to further reduce system-wide energy consumption and decrease peripheral latency by recognizing the completion of the voltage-independent state and thus completing the operation early.

Peripheral operations are specified for a worst-case duration by the manufacturer that may depend on a number of factors including age and temperature. Most peripheral devices provide a mechanism for signaling that operations completed earlier than the maximum. ACR develops adaptive timing, current usage and charge consumption heuristics for estimating early completion of peripheral operations.

The estimate is verified upon returning from the voltage-independent state and the heuristic is updated with the results. In this fashion, the algorithms are resistant to variations in behavior that may occur across the lifecycle of the device. ACR is measured against a variety of embedded peripherals and is shown to further decrease peripheral energy consumption decrease peripheral latency with minimal computational overhead.

For example, when writing a page of EEPROM a voltage-independent wait state is encountered that is specified to a maximum duration of 5ms. However, that specification is for the worst case and is more suitable for a timeout value. The current consumption profile of an EEPROM write operation at varying voltages is shown in Figure 26.

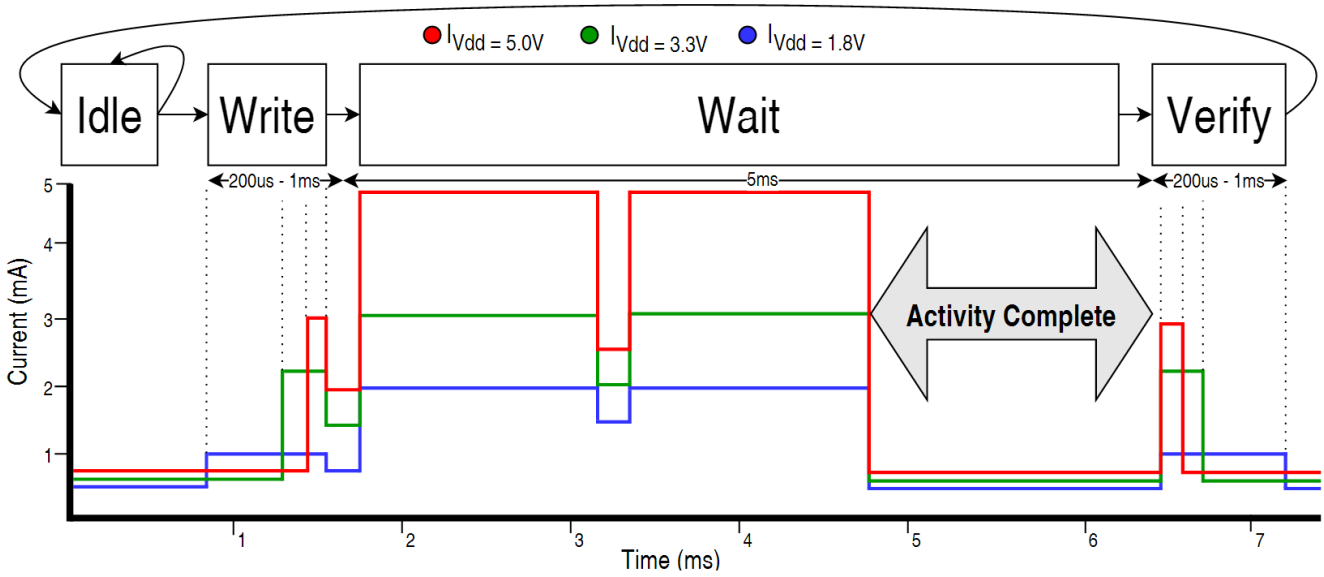


Figure 26: EEPROM Write Current Profile

As it transitions through the Idle → Write → Wait → Verify states, it can be inferred from the current profile that the operation completed by the 5ms mark and that it was not necessary to delay until approximately 6.5ms per the specification. In the case of EEPROM and most peripheral devices, there is a register that can be polled and it indicates when the write has completed. Polling this register requires the MCU to communicate with the peripheral and thus results in transitioning to a voltage-dependent state.

Thus, accurate estimations can decrease latency and energy consumption, but inaccurate estimates can result in an early transition to a voltage-dependent state and thus increase energy consumption.

There are a wide variety of peripheral devices with a correspondingly wide variety of completion determinism and current profiles. Devices with highly deterministic timing respond best to the timing heuristic while those with variable timing respond best to current or charge heuristics.

ACR seeks to estimate and detect early completion of operations in peripheral devices by applying timing and current usage heuristics. Through early completion detection, ACR is able to decrease both latency and system-wide energy consumption. ACR is particularly advantageous to systems implementing IODVS by decreasing the effective duration of voltage-independent states.

Related Work

Timing Heuristic

Peripheral operations can vary in their latency or completion times due to a number of factors. Temperature can significantly affect the completion time for peripherals with fairly deterministic timing requirements such as DRAM [32]. Device aging can also affect timing due to a number of issues resulting from fundamental semiconductor physics [33]. Furthermore, some devices simply have non-deterministic completion times due to features such as MMUs and caches that are implemented in various data storage devices like Micro-SD cards, or age and wear as they effect FLASH storage timing.

Because the latency can vary significantly between operations, it is necessary to develop a timing heuristic that can adapt to slowly changing effects like age and temperature as well as rapidly changing factors like cache hits and misses. Adaptive delay estimation is not a new problem [34] and research continues to compensate for non-deterministic delay with different approaches for wireless communications, control systems and mass storage latency [35].

Current Heuristic

The completion of some peripheral operations are easily detectable by their current consumption profile. These devices have a distinct and deterministic current profile that can be characterized and used to estimate the moment when an operation completes.

Simple and differential power analysis (SPA and DPA) attacks are performed by monitoring device current consumption with very fine grained detail. These attacks seek to undermine encryption techniques by monitoring the current consumption of the processor and detecting the moment at which the processor executes a branch operation [36]. The attacks have been performed on an ARM Cortex MCU using AES and required an extensive measurement setup to accomplish [37]. ACR is inspired by this previous work using fine-grained in-circuit current measurement and fortunately benefits from much more lenient sampling requirements.

Charge Heuristic

For devices with highly variable timing and dynamic current consumption characteristics, integrating the current consumption of the device throughout an operation can allow for better detection of completion. Some operations can be characterized by the amount of charge necessary to complete them. This technique is referred to as “coulomb counting” and is a common technique used to determine the state of charge in rechargeable batteries [38].

Methods and Materials

Development Platform

ACR and IODVS are hosted on a STM32F427 MCU implemented on the Discovery evaluation board. The board provides 64MB of SDRAM which allows for simultaneous sampling throughout the test suite at very high speed. All experiments were sampled at 1MSPS and the SDRAM allowed any individual experiment so last up to 1 full second.

The timing requirements of SDRAM are less strict than asynchronous SRAM and so tests were performed to ensure that memory bandwidth would be sufficient for the application. ADCs were driven using a timer and their results were transmitted via DMA to the external SDRAM chip along with device state information at a rate of 1MSPS.

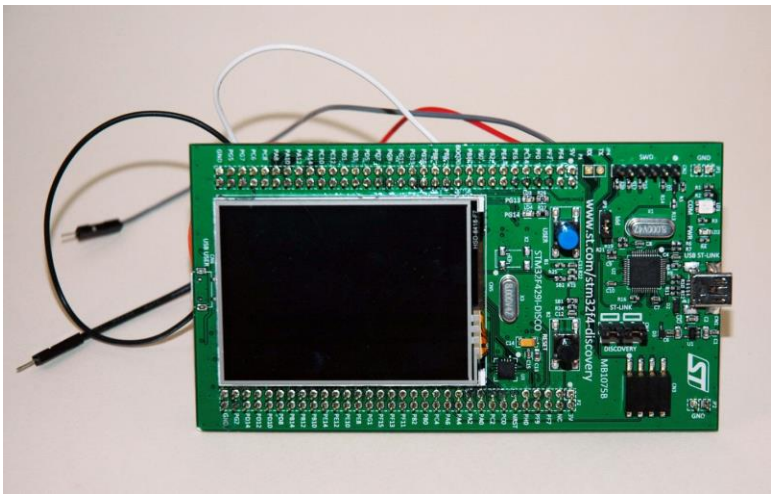


Figure 27: STM32F429 Discovery Front

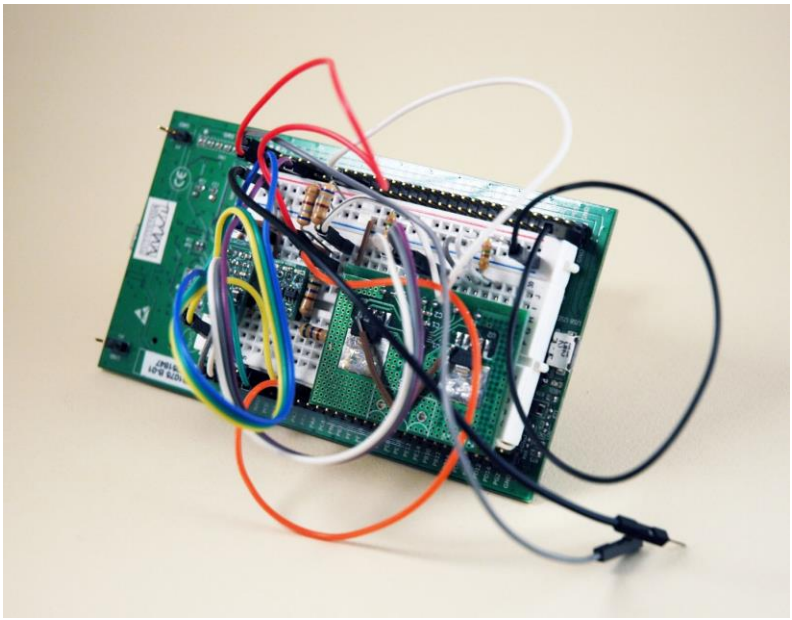


Figure 28: STM32F429 Discovery Back

Adjustable Step-Down Module with Feedback (ASDM-300F)

Previous implementations used an adjustable SMPS to provide power to the peripheral devices. While this device is very efficient, it also produces a significant amount of noise. The noise does not affect operation of the device, but could lead to a misinterpretation of the analog measurements. This problem is addressed through the development of a module that incorporates an adjustable SMPS, the current measurement circuitry and an onboard LDO for noise reduction. The module exposes 8 pins which are compatible with a DIP-8 standard package configuration.

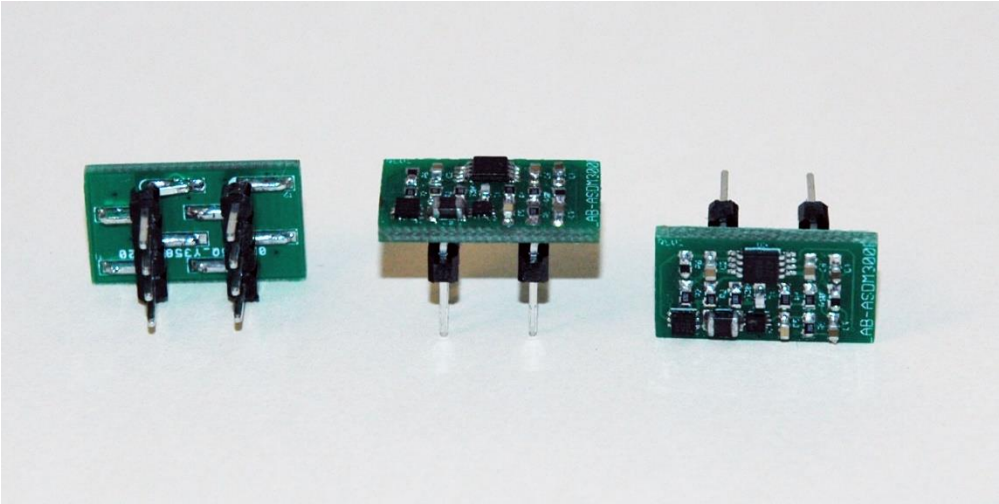


Figure 29: ASDM-300F

Results

ACR was tested against seven peripherals and …

Conclusion

CHAPTER 5: SUPERVISED IODVS

IODVS has been shown to considerably reduce energy consumption in embedded peripherals. Throughout the course of previous work, the problem of interfering voltage changes was encountered. That is, where two or more peripherals coexist on the same voltage domain and one peripheral is more tolerant of voltage changes than others.

For example, the EEPROM in previous experiments was capable of operating at 1.8V while the SDCard on the same domain would undergo a reset condition if the domain voltage were switched temporarily to 1.8V. Not only would the SDCard need to undergo a lengthy reset procedure, but the device driver expects the SDCard to be in an operational state when the next device access is issued. It is unlikely that the device driver would be able to handle the condition. Designing the device driver to handle random state changes outside of its control would result in a very inefficient driver.

A number of options are available to address the problem of domain voltage interference. The trivial solution would be to put each device on its own individual voltage domain. Of course, implementing a voltage domain for each peripheral in an embedded system would be both cost and size prohibitive. Additionally, this method inevitably operates an SMPS in a very inefficient voltage translation region (the very lightly loaded region).

A voltage supervisor implemented at the driver or OS level is a natural fit for this type of problem. A simple mitigating option is to notify the drivers of all devices on a domain about voltage changes that are taking place. If two devices are on the same voltage domain and an IODVS voltage change is requested, then the other device driver is notified of the change. In this way, at least the device and driver can maintain consistency and the potential for devastating faults is reduced.

In fact, the supervisor opens up a number of options for mitigating inter-device interference. By registering the peripheral power profile with the supervisor, individual drivers can provide feedback to the supervisor as to how a voltage change would affect the device. If a device on the voltage domain would be affected by the requested change, then that change would be vetoed by the supervisor. Furthermore, rather than a simplistic binary decision, a temporal cost is evaluated against each voltage change. That is, for instance the time required to reinitialize an SDCard after a voltage change that causes a reset.

Ultimately, usage statistics like those used in DPM implementations are investigated in order to both optimize energy consumption and minimize response time. These usage statistics are contrasted against the temporal cost of voltage changes on the domain.

A new hardware platform was developed for the exploration of supervised IODVS. The Precise Real-time In-circuit Monitoring and Energy management system (PRIME) provides high resolution, fine-grained control and a wide variety of peripheral devices for experimentation. The remainder of this chapter discussed the hardware developed to investigate Supervised IODVS at a fine-grained level, as well as the results of various supervisory policies.

Peripheral Power Switch

Another limitation of previous experiments is that current measurements were taken at the supply of the voltage domain. While conducting experiments on one device, the effect of that voltage change was actually seen as to how it affected every device on the domain. This issue is addressed through the development of the Peripheral Power Switch (3x Circuits, 3.0Amp Max, Dip Package) – PPS-330D.

The PPS-330D is 8-Pin DIP package compatible and provides inputs for 3 voltages and ground. A peripheral device or group of peripheral devices are connected to the output voltage and the output ground. The remaining two pins are used to select between which of the three domains (or disconnection) that the device is connected to. Selection is done using a 2→4 decoder and the selected domain is indicated via LEDs on the module.

In addition to load isolation, the device provides another option for load management. Given that a peripheral device may operate via IODVS in the range of 1.8→3.3V, some domains may already be at the requested voltage. Therefore, rather than adjusting a voltage domain, the device could simply be switched to a more compatible domain for its current mode of operation.

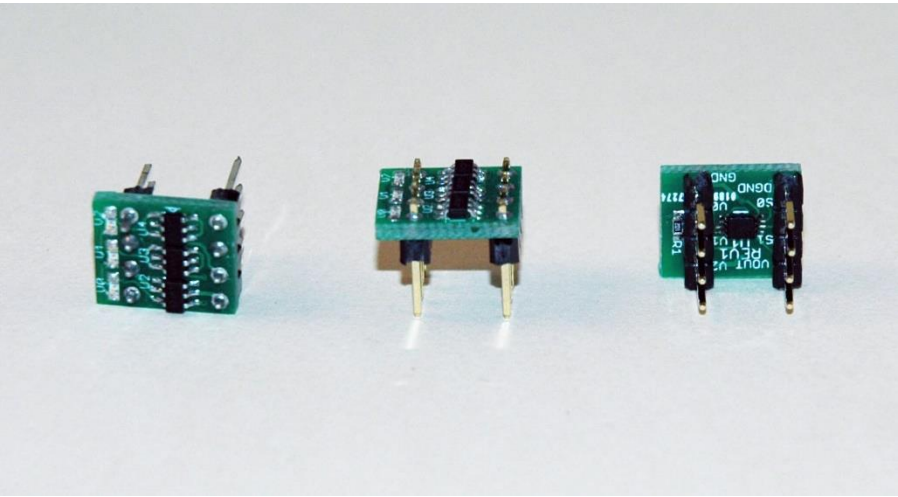


Figure 30: PPS-330D

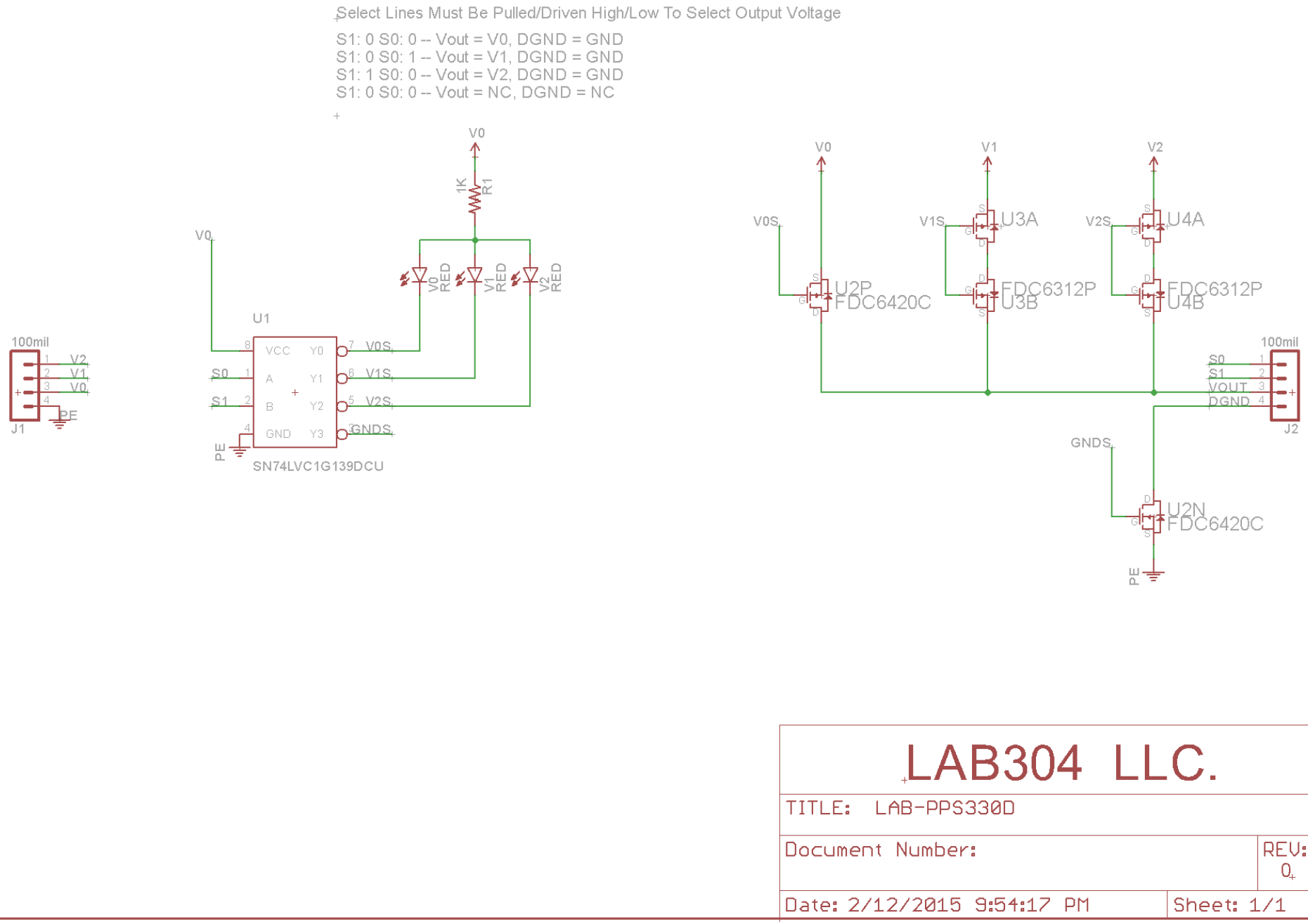


Figure 31: PPS-330D Schematic

Programmable Load Regulator

Through IODVS the current consumption requirements of various operations on multiple different devices was characterized. The PLR-5010D (Programmable Load Regulator, 5V, 1.0A, Dual Output) allows

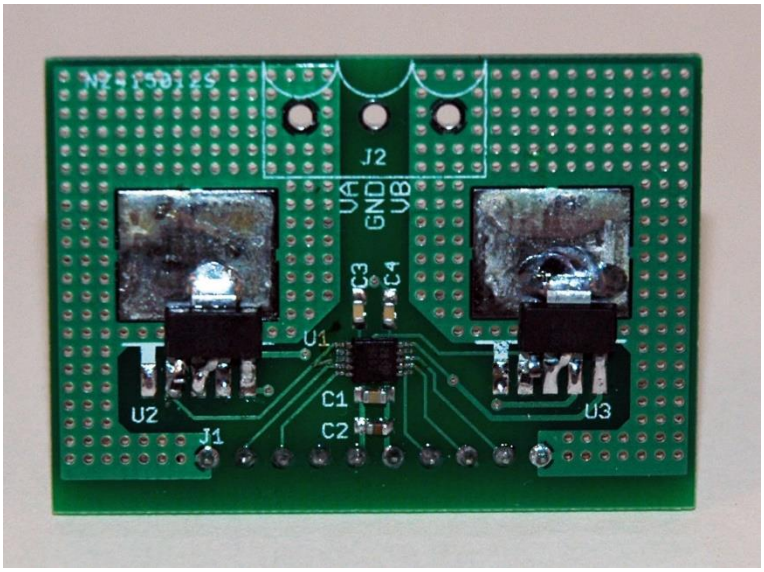


Figure 32: PLR-5010D

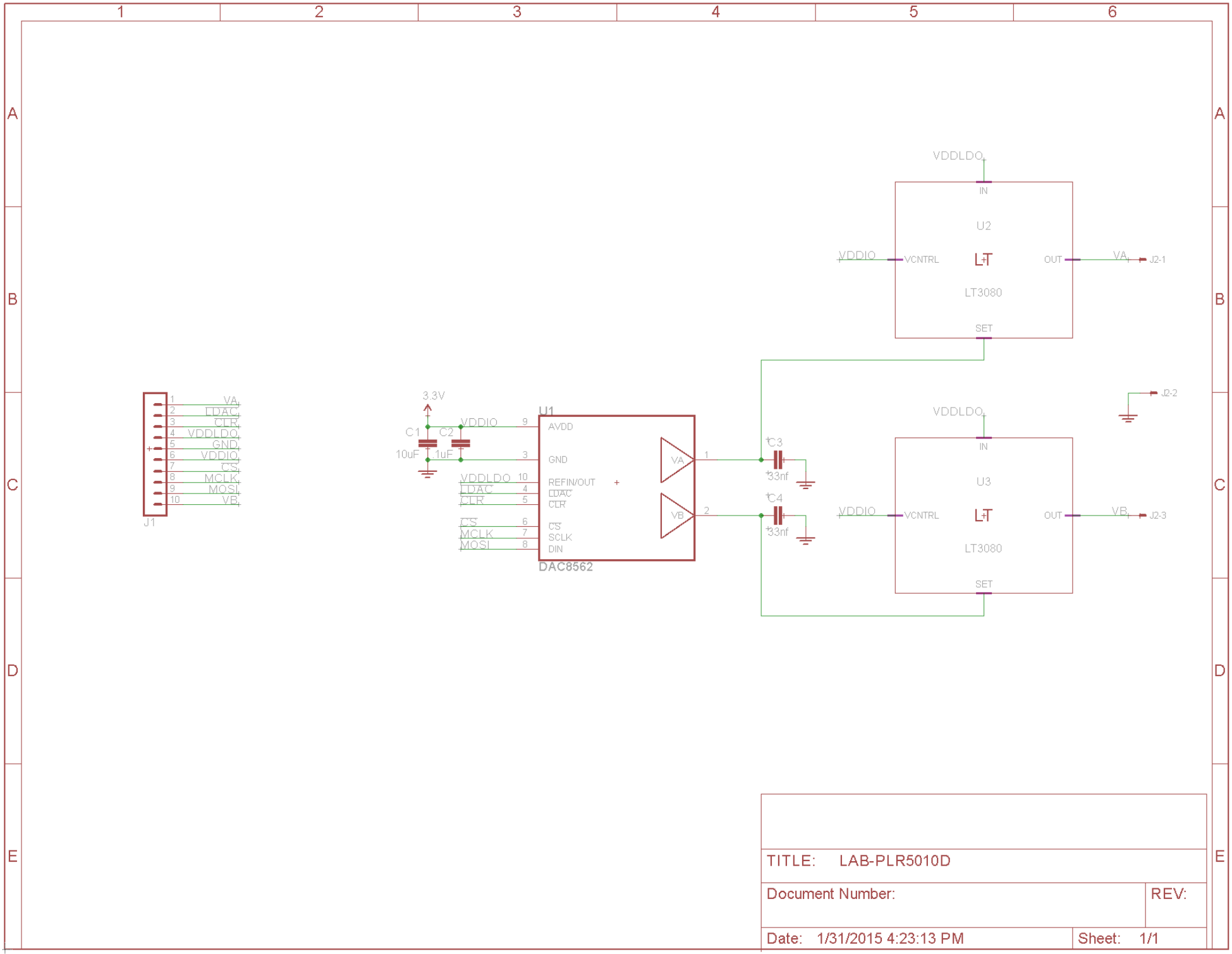
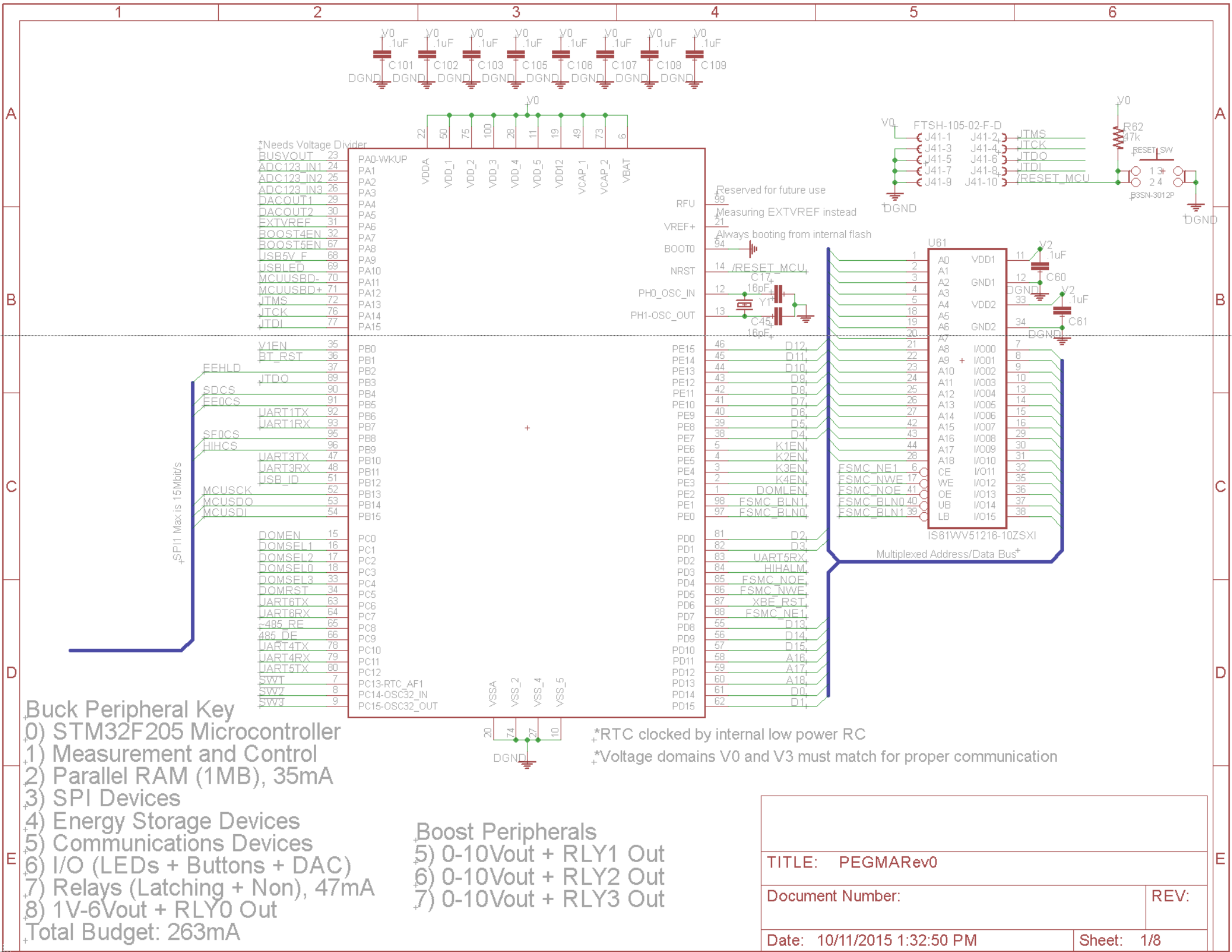


Figure 33: PLR5010-D Schematic

APPENDIX A

PEGMA Schematic

1. Microcontroller pinout and SRAM connection
2. Renewable input boost circuitry, measurement and modulation
3. Energy storage and peripheral boost circuitry
4. Stepdown power supplies (peripheral domains)
5. Peripheral domain current measurement
6. Peripherals under test
7. Communications peripherals
8. Analog domain



*RTC clocked by internal low power RC

*Voltage domains V0 and V3 must match for proper communication

Buck Peripheral Key

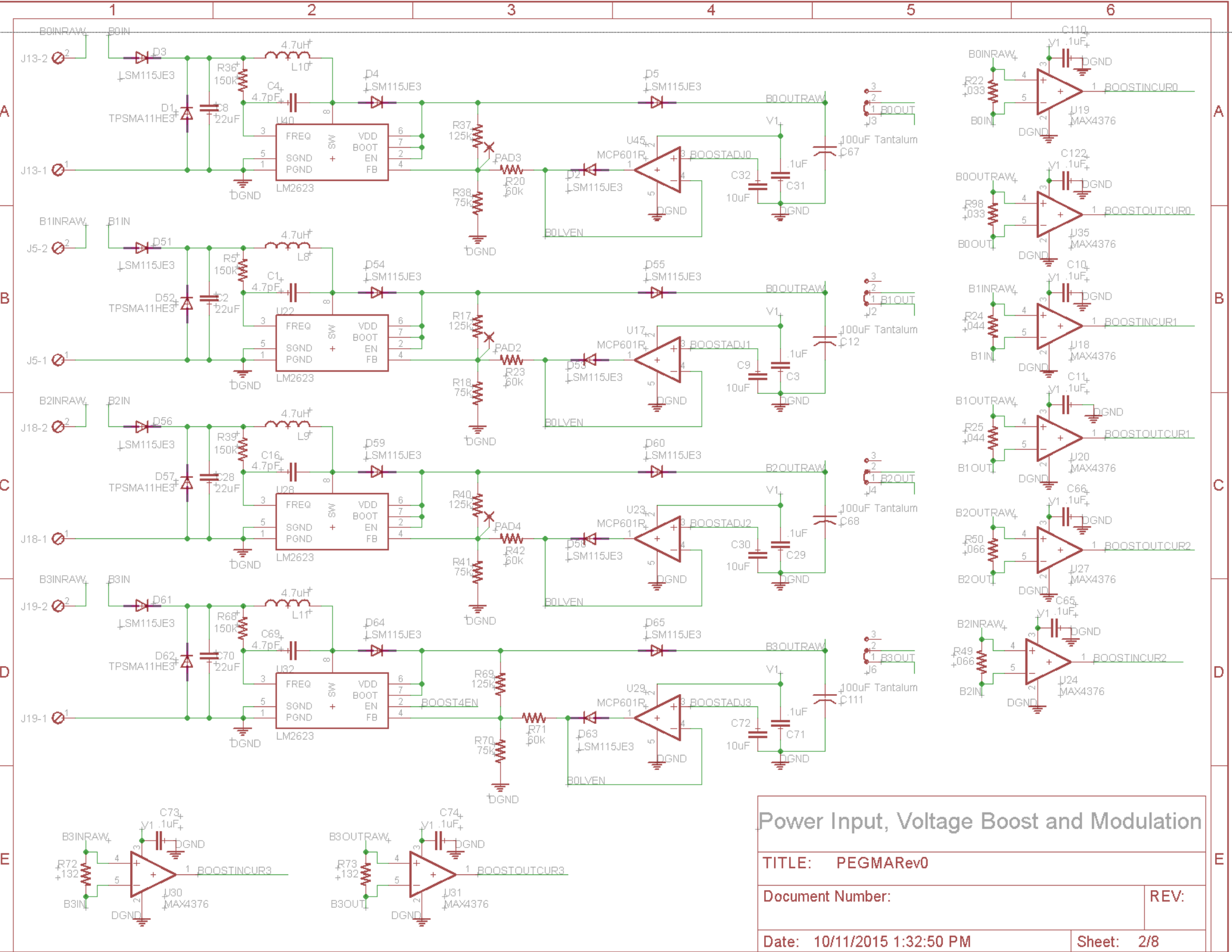
0) STM32F205 Microcontroller
1) Measurement and Control
2) Parallel RAM (1MB), 35mA
3) SPI Devices
4) Energy Storage Devices
5) Communications Devices
6) I/O (LEDs + Buttons + DAC)
7) Relays (Latching + Non), 47mA
8) 1V-6Vout + RLY0 Out

Total Budget: 263mA

Boost Peripherals

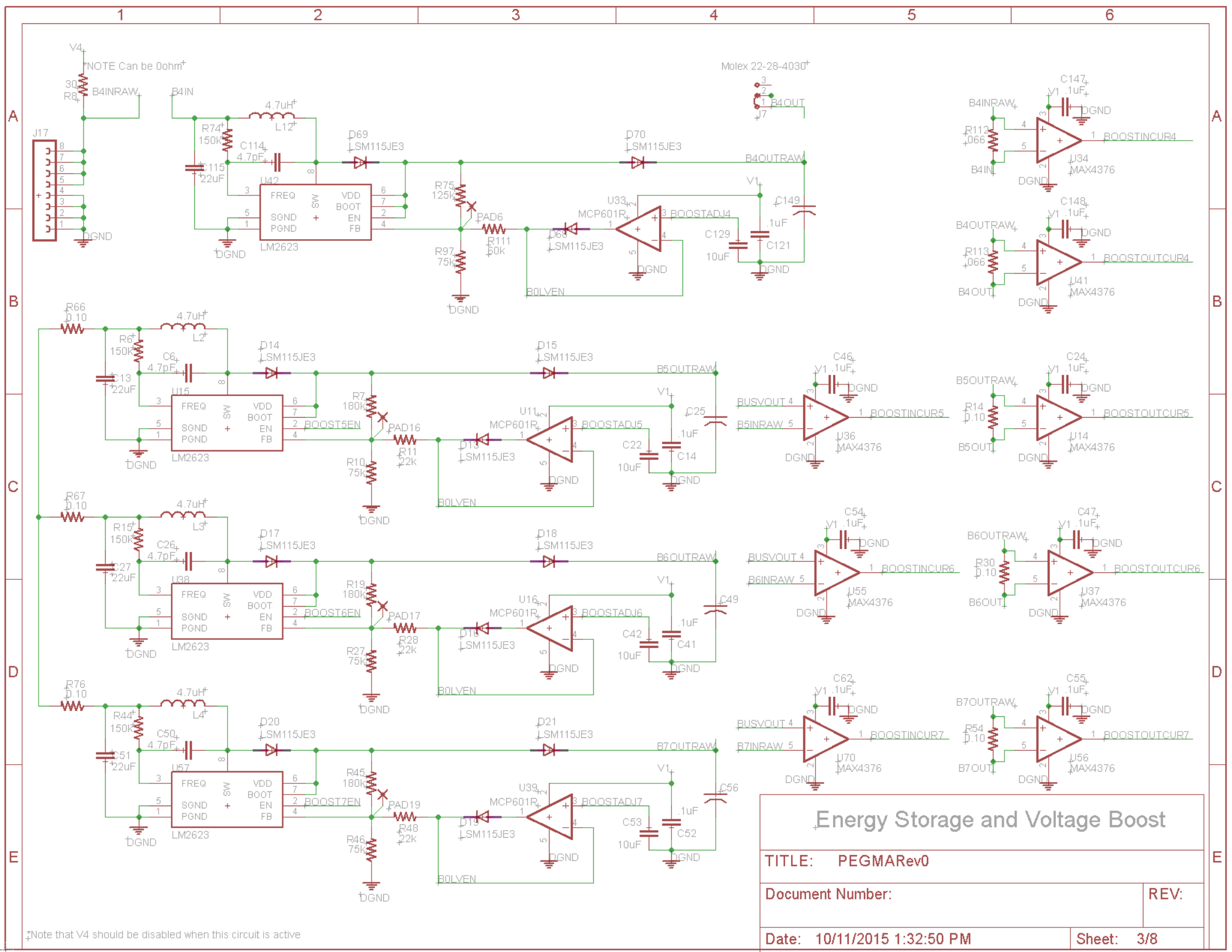
5) 0-10Vout + RLY1 Out
6) 0-10Vout + RLY2 Out
7) 0-10Vout + RLY3 Out

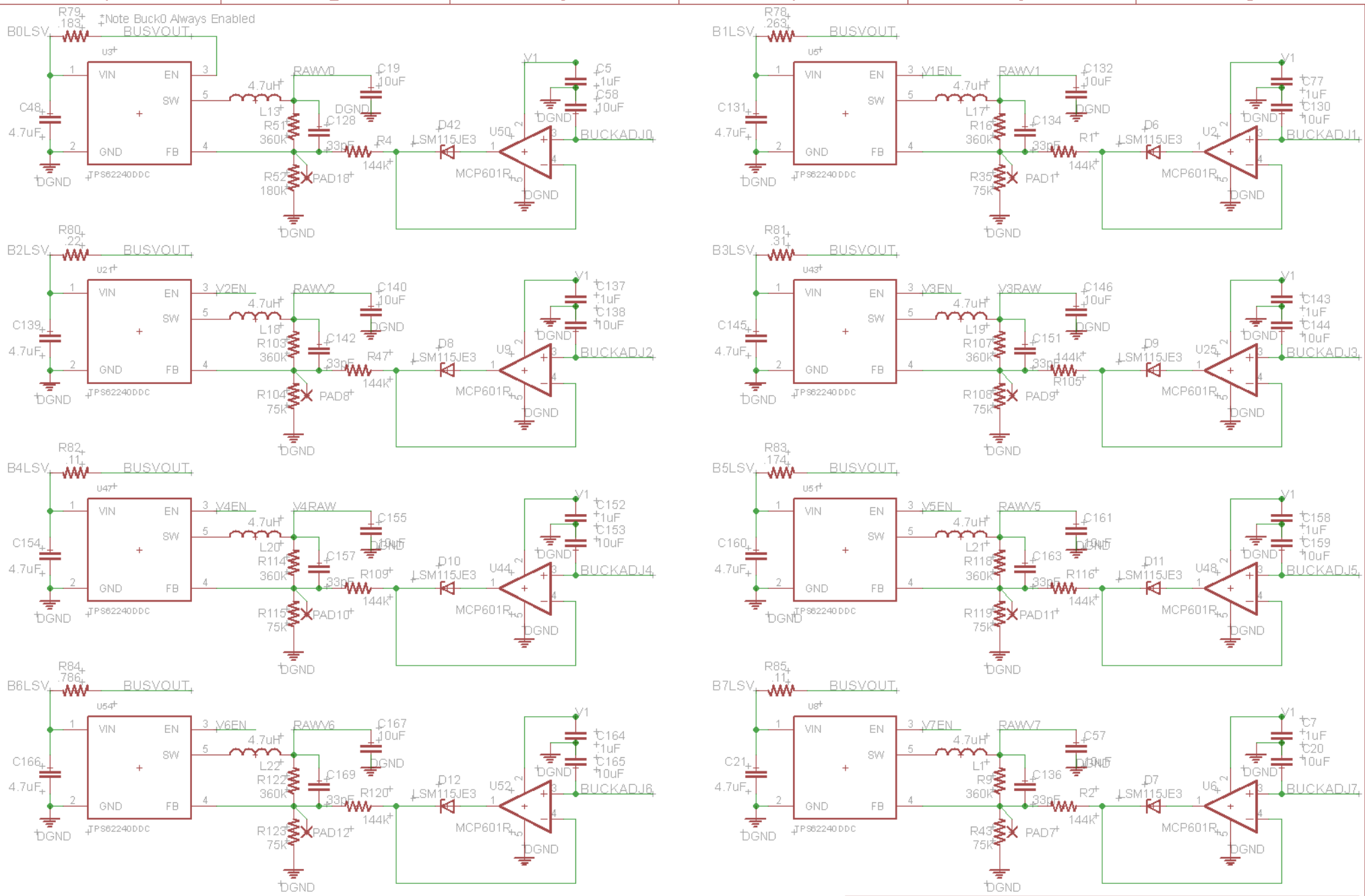
TITLE: PEGMARev0	
Document Number:	REV:
Date: 10/11/2015 1:32:50 PM	Sheet: 1/8



Power Input, Voltage Boost and Modulation

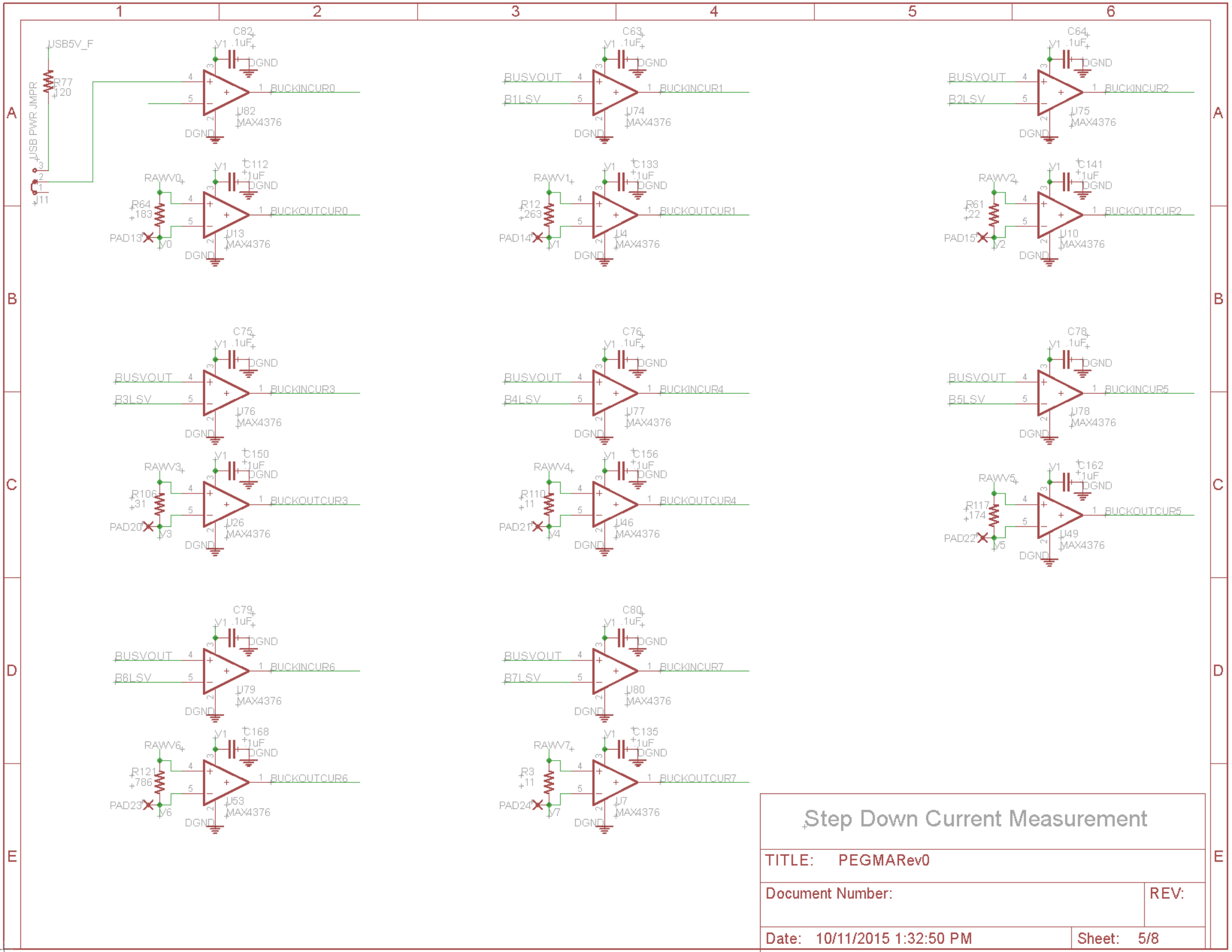
TITLE:	PEGMARev0
Document Number:	REV:
Date: 10/11/2015 1:32:50 PM	Sheet: 2/8

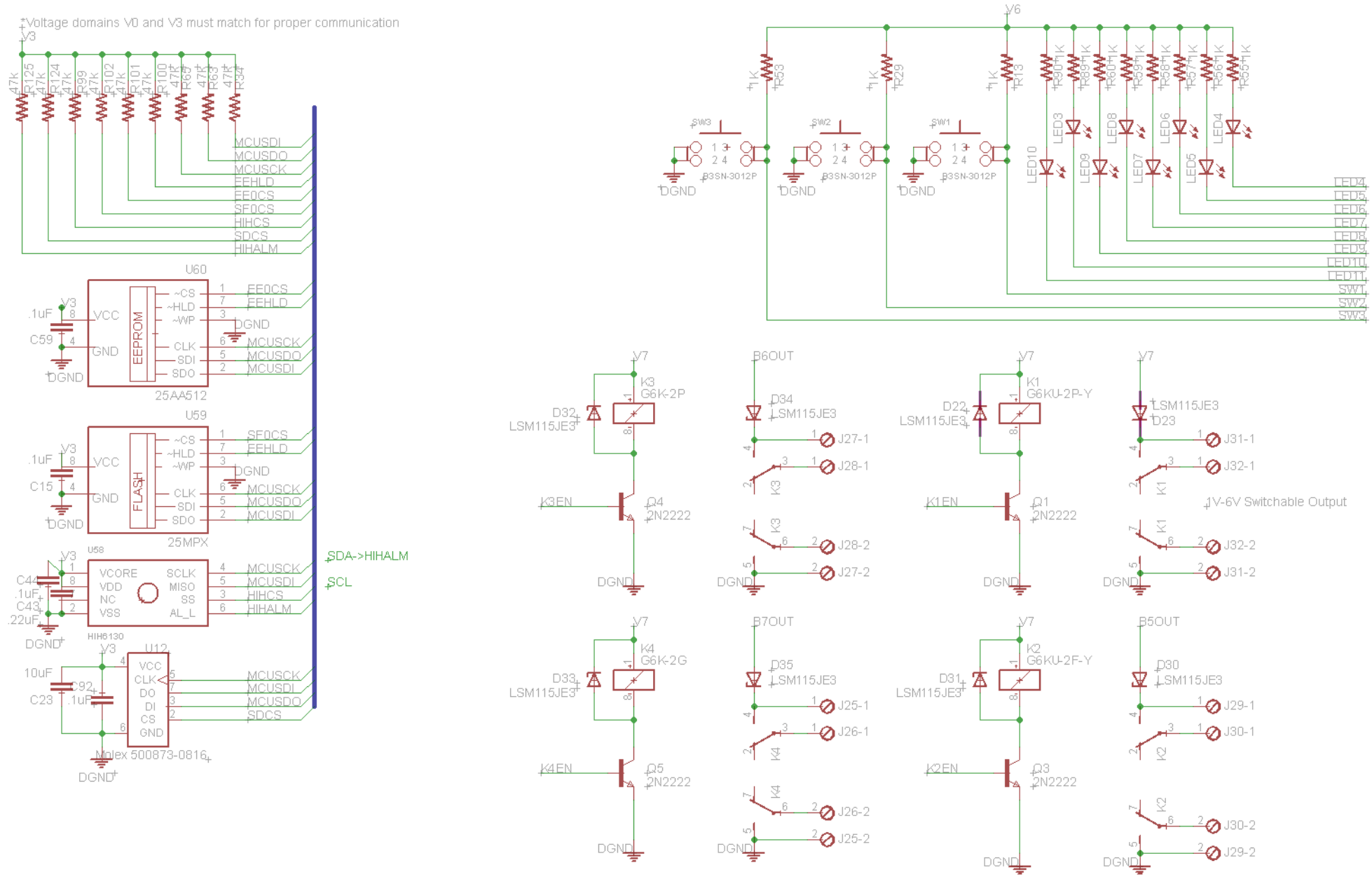

NOTE Can be 0ohm
Molex 22-28-4030
B4INRAW
B4IN
B4OUT
B4OUTRAW
J17
J7
R8
30
4.7uH
L12
R74
150k
C114
4.7pF
C115
22uF
U42
FREQ
SGND
PGND
SW
VDD
BOOT
EN
FB
LM2623
D69
LSM115JE3
D70
LSM115JE3
R75
125k
PAD6
R111
60k
R97
75k
U33
MCP601R
BOOSTADJ4
D68
LSM115JE3
C129
10uF
C121
.1uF
C149
V1
B0LVEN
DGND
C147
C148
R112
066
R113
066
BOOSTINCUR4
BOOSTOUTCUR4
U34
U41
MAX4376
R66
0.10
R6
150k
C6
4.7pF
C13
22uF
L2
U15
D14
LSM115JE3
D15
LSM115JE3
B5OUTRAW
R7
180k
PAD16
R11
22k
R10
75k
BOOST5EN
U11
BOOSTADJ5
D13
LSM115JE3
C22
10uF
C14
C25
C46
C24
BUSVOUT
B5INRAW
R14
0.10
BOOSTINCUR5
BOOSTOUTCUR5
U36
U14
R67
0.10
R15
150k
C26
4.7pF
C27
22uF
L3
U38
D17
LSM115JE3
D18
LSM115JE3
B6OUTRAW
R19
180k
PAD17
R28
22k
R27
75k
BOOST6EN
U16
BOOSTADJ6
D16
LSM115JE3
C42
10uF
C41
C49
C54
C47
B6INRAW
R30
0.10
BOOSTINCUR6
BOOSTOUTCUR6
U55
U37
R76
0.10
R44
150k
C50
4.7pF
C51
22uF
L4
U57
D20
LSM115JE3
D21
LSM115JE3
B7OUTRAW
R45
180k
PAD19
R48
22k
R46
75k
BOOST7EN
U39
BOOSTADJ7
D19
LSM115JE3
C53
10uF
C52
C56
C62
C55
B7INRAW
R54
0.10
BOOSTINCUR7
BOOSTOUTCUR7
U70
U56
MAX4376
Energy Storage and Voltage Boost
TITLE: PEGMARev0
Document Number:
REV:
Date: 10/11/2015 1:32:50 PM
Sheet: 3/8
Note that V4 should be disabled when this circuit is active



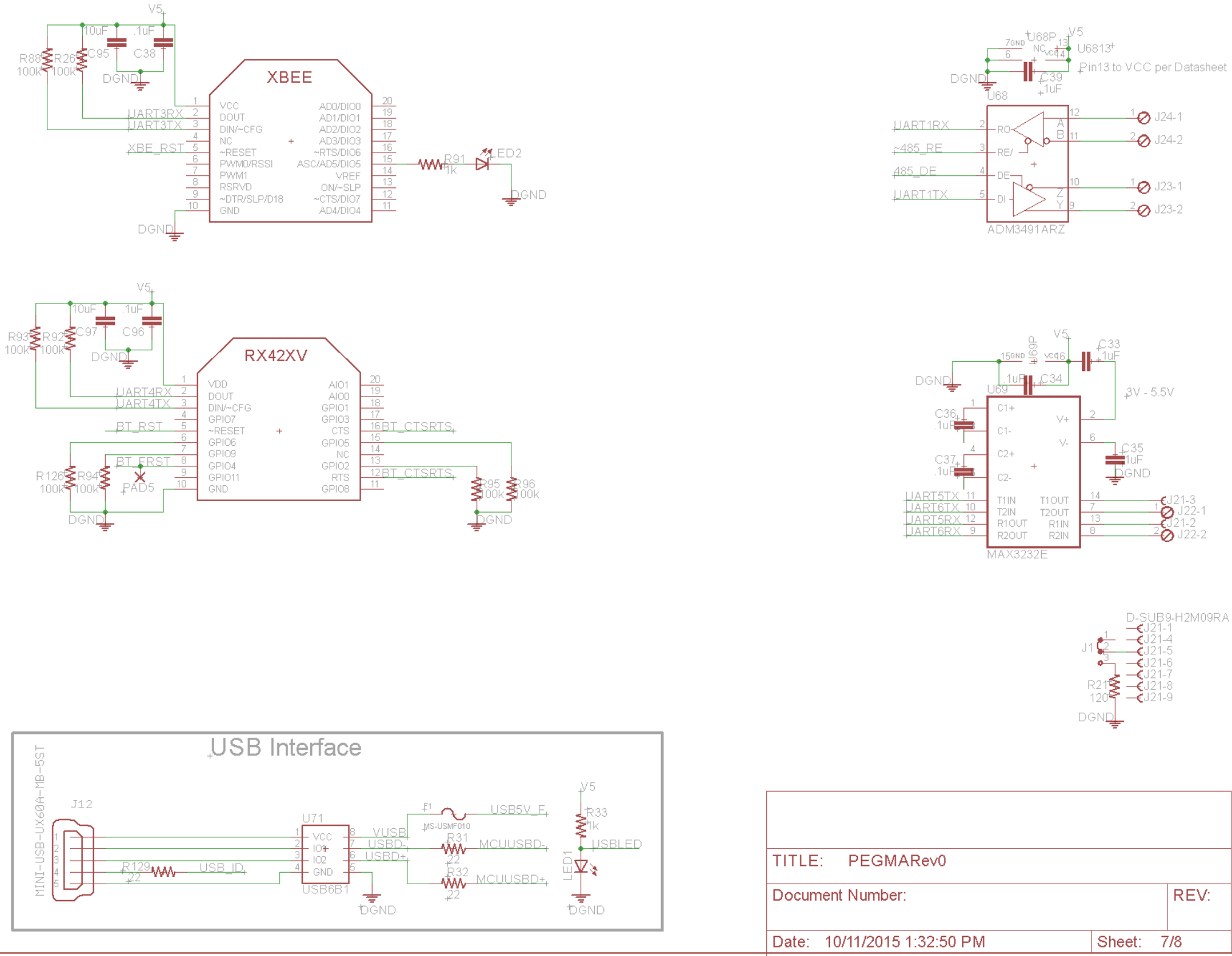
Voltage Step Down and Modulation

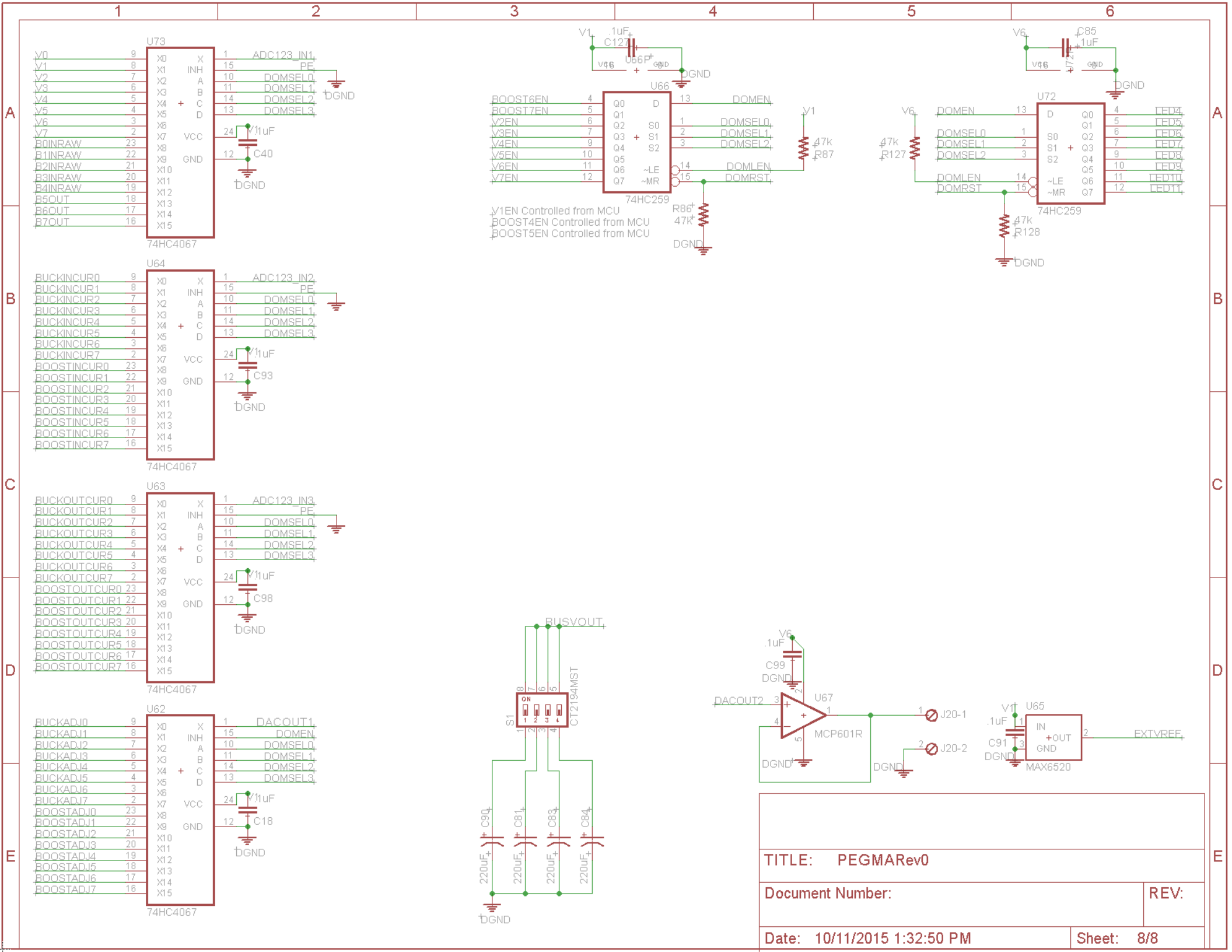
TITLE: PEGMARev0	
Document Number:	REV:
Date: 10/11/2015 1:32:50 PM	Sheet: 4/8


1
2
3
4
5
6
A
B
C
D
E
USB5V_F
R77
+20
USB PWR JMPR
+11
C82
.1uF
DGND
U82
MAX4376
BUCKINCUR0
BUCKOUTCUR0
RAWV0
R64
+183
PAD13
V0
C112
+1uF
U13
C63
U74
BUCKINCUR1
BUCKOUTCUR1
BUSVOUT
B1LSV
RAWV1
R12
+263
PAD14
V1
C133
C64
U75
BUCKINCUR2
BUCKOUTCUR2
B2LSV
RAWV2
R61
+22
PAD15
V2
C141
C75
U76
BUCKINCUR3
BUCKOUTCUR3
B3LSV
RAWV3
R106
+31
PAD20
V3
C150
C76
U77
BUCKINCUR4
BUCKOUTCUR4
B4LSV
RAWV4
R110
+11
PAD21
V4
C156
C78
U78
BUCKINCUR5
BUCKOUTCUR5
B5LSV
RAWV5
R117
+174
PAD22
V5
C162
C79
U79
BUCKINCUR6
BUCKOUTCUR6
B6LSV
RAWV6
R121
+786
PAD23
V6
C168
C80
U80
BUCKINCUR7
BUCKOUTCUR7
B7LSV
RAWV7
R3
+11
PAD24
V7
C135
Step Down Current Measurement
TITLE: PEGMARev0
Document Number:
REV:
Date: 10/11/2015 1:32:50 PM
Sheet: 5/8

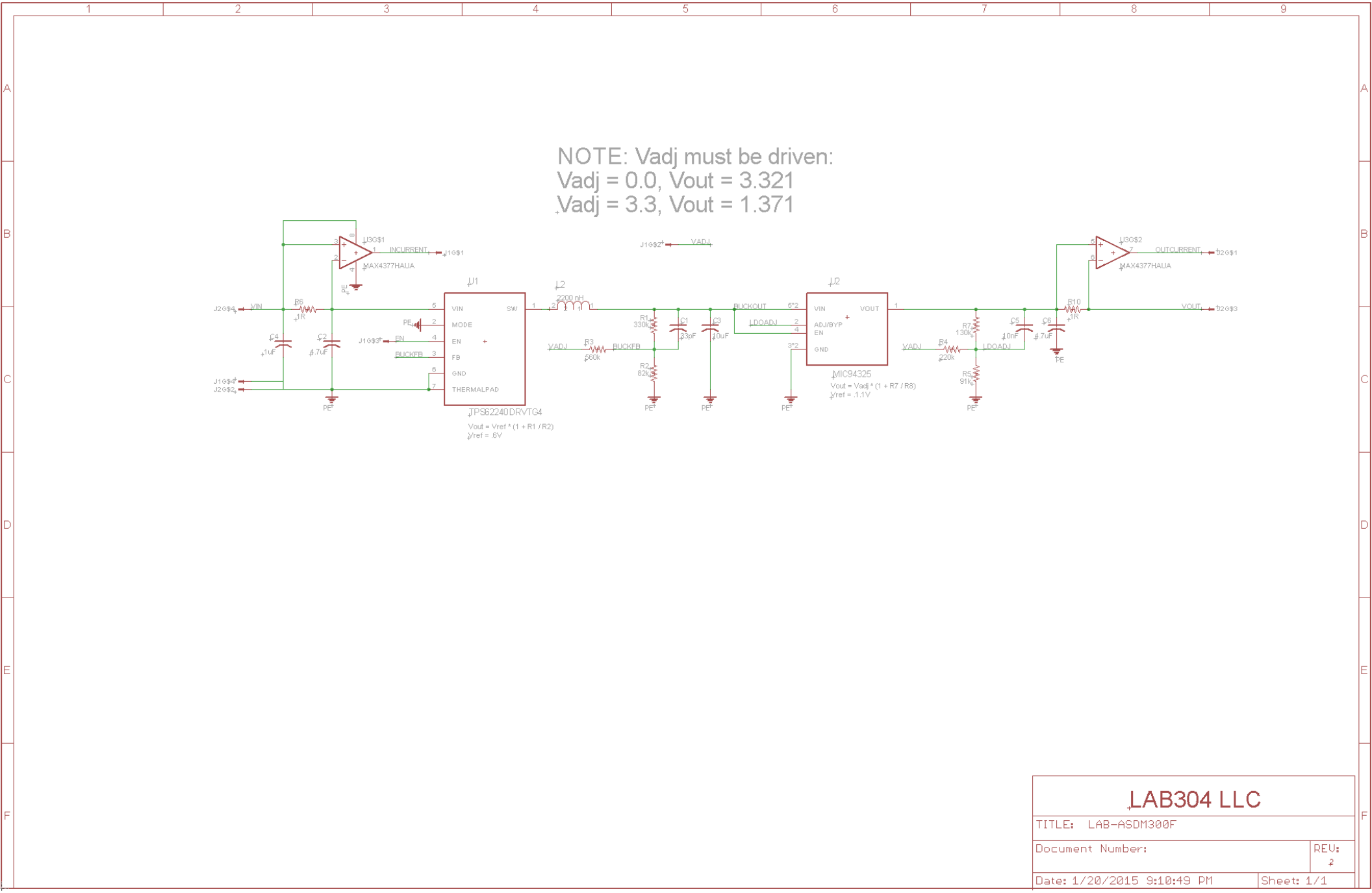


TITLE: PEGMARev0		
Document Number:		REV:
Date: 10/11/2015 1:32:50 PM	Sheet: 6/8	


XBEE
RX42XV
ADM3491ARZ
MAX3232E
D-SUB9-H2M09RA
USB Interface
MINI-USB-UX60A-MB-5ST
USB6B1
Pin13 to VCC per Datasheet
TITLE: PEGMARev0
Document Number:
REV:
Date: 10/11/2015 1:32:50 PM
Sheet: 7/8


U73
V0
V1
V2
V3
V4
V5
V6
V7
B0INRAW
B1INRAW
B2INRAW
B3INRAW
B4INRAW
B5OUT
B6OUT
B7OUT
X0
X1
X2
X3
X4
X5
X6
X7
X8
X9
X10
X11
X12
X13
X14
X15
X
INH
A
B
C
D
VCC
GND
74HC4067
ADC123_IN1
PE
DOMSEL0
DOMSEL1
DOMSEL2
DOMSEL3
1uF
C40
DGND
U64
BUCKINCUR0
BUCKINCUR1
BUCKINCUR2
BUCKINCUR3
BUCKINCUR4
BUCKINCUR5
BUCKINCUR6
BUCKINCUR7
BOOSTINCUR0
BOOSTINCUR1
BOOSTINCUR2
BOOSTINCUR3
BOOSTINCUR4
BOOSTINCUR5
BOOSTINCUR6
BOOSTINCUR7
ADC123_IN2
C93
U63
BUCKOUTCUR0
BUCKOUTCUR1
BUCKOUTCUR2
BUCKOUTCUR3
BUCKOUTCUR4
BUCKOUTCUR5
BUCKOUTCUR6
BUCKOUTCUR7
BOOSTOUTCUR0
BOOSTOUTCUR1
BOOSTOUTCUR2
BOOSTOUTCUR3
BOOSTOUTCUR4
BOOSTOUTCUR5
BOOSTOUTCUR6
BOOSTOUTCUR7
ADC123_IN3
C98
U62
BUCKADJ0
BUCKADJ1
BUCKADJ2
BUCKADJ3
BUCKADJ4
BUCKADJ5
BUCKADJ6
BUCKADJ7
BOOSTADJ0
BOOSTADJ1
BOOSTADJ2
BOOSTADJ3
BOOSTADJ4
BOOSTADJ5
BOOSTADJ6
BOOSTADJ7
DACOUT1
C18
V1
.1uF
C127
U66P
U66
BOOST6EN
BOOST7EN
V2EN
V3EN
V4EN
V5EN
V6EN
V7EN
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
D
S0
S1
S2
~LE
~MR
74HC259
DOMEN
DOMLEN
DOMRST
R87
47k
R86
V1EN Controlled from MCU
BOOST4EN Controlled from MCU
BOOST5EN Controlled from MCU
V6
C85
U72
R127
R128
LED4
LED5
LED6
LED7
LED8
LED9
LED10
LED11
RUSVOUT
S1
CT2194MST
ON
C90
C81
C83
C84
220uF
U67
DACOUT2
MCP601R
C99
J20-1
J20-2
U65
C91
IN
+OUT
GND
MAX6520
EXTVREF
TITLE: PEGMARev0
Document Number:
REV:
Date: 10/11/2015 1:32:50 PM
Sheet: 8/8

APPENDIX B: ASDM-300F SCHEMATIC



LAB304 LLC	
TITLE: LAB-ASDM300F	
Document Number:	REV: 2
Date: 1/20/2015 9:10:49 PM	Sheet: 1/1