Using MRAM to Optimize System Energy Consumption

By Duncan Bennett, Product Marketing Manager, Everspin Technologies, Inc.

For many wireless and portable applications, especially in the growing Internet of Things applications, there is a critical energy budget (total power consumed over time). As a member of the technical staff at Everspin Technologies I spend a lot of time working with system designers on applications for MRAM. An idea that I have wanted to investigate for a long time is whether the fast-Write and power-up-to-Write times for MRAM can significantly reduce total system energy consumption compared to either EEPROM or Flash.

Abstract

For this article, I compared system energy consumption in a typical data acquisition system using Flash, EEPROM or MRAM. The comparisons showed that:

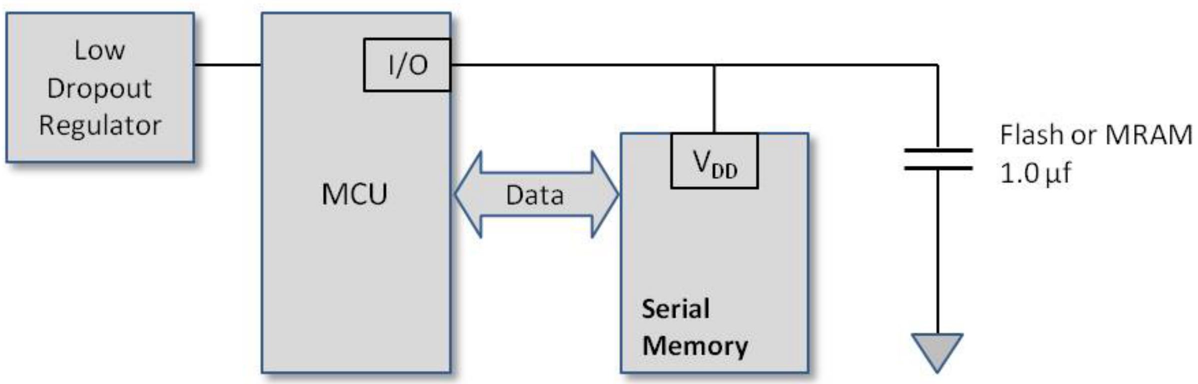
1. Non-volatile memory write time is a major contributor to overall system energy consumption, thus the shorter write time of MRAM can actually reduce total energy consumption.
2. Further system energy reductions can be achieved using a power-gating architecture with MRAM because its faster power-up to write time allows MRAM standby power to be reduced to zero.

Typical System

I used the schematic in **Figure 1** to represent the Low-Voltage Dropout Regulator (LDO), microcontroller (MCU), non-volatile memory and a decoupling capacitor typical in data acquisition applications such as medical monitors, data loggers, etc. Other system components, such as sensors and the power they consume were not considered.

The MCU was assumed to be in a low power sleep state and with a periodic wake up to make an acquisition of data. The data acquired was stored in the non-volatile memory and then the system returned to the sleep state.

Figure 1



Application with Flash or MRAM Using Power Gating for Standby Mode

I compared non-volatile memory with a SPI interface and looked at only Write operations, which typically consume much more power than Read operations. The number of data bytes that can be written is four less than the number of bytes on the SPI bus due to overhead of the Write command, the WREN bit and two address bytes. The number of bytes written to the non-volatile memory was selected as 4 and 46. Four is perhaps most likely, representing the storage of one data acquisition sample. Forty-six is the optimum amount of data that can be written to an MRAM when powered from a 1.0 µF decoupling capacitor.

Power Gating Considerations

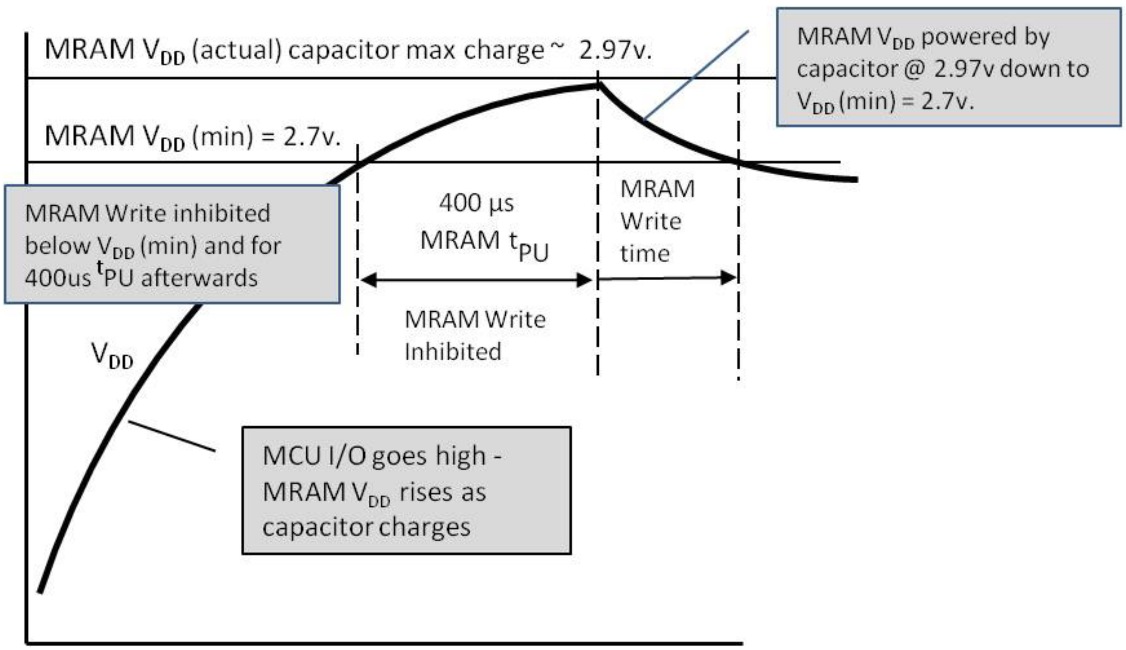
A few quick calculations revealed that the decoupling capacitor is very important when power gating. The energy used to charge the capacitor from zero is significant.

EEPROM can be powered directly from the I/O of a standard microcontroller, typically 4mA, and consequently a small 0.1µF capacitor was used for decoupling. MRAM and Flash need more current than is available from a standard MCU I/O. For these memories I assumed a larger decoupling capacitor so that the Flash or MRAM could run on the energy stored in the larger capacitor.

Phases of the WRITE operation

The energy consumption of the non-volatile memory was calculated during the phases of the write operation shown in **Figure 2**.

Figure 2



MRAM V_{DD} During Power-Up and Access Times using Power Gating

Rise Time

During this phase I assumed that all of the energy goes into the decoupling capacitor and that the non-volatile memory consumes negligible energy.

Power-Up Time

Once the voltage on V_{DD} is above a threshold, a small delay (^{t}PU) is required for MRAM to become ready but not for EEPROM or Flash. During this phase I assumed that the MRAM consumes the current shown in the stand-by specification of the datasheet.

Write Time

During this phase, the non-volatile memory consumes the current shown in the active specification of the datasheet.

Assuming a 3.3V system with a tolerance of ± 10%, the lowest voltage on the I/O could be 3.3V -10% = 2.97V. This voltage of 2.97V was used in calculations.

Energy Calculations

I started by looking at the energy used by the non-volatile memory during writing.

Energy Used by an EEPROM

I looked at a typical 3.3V EEPROM with a standby current of 1µA, a write time of 5ms and a write current of 3mA.

For EEPROM I assumed the following:

1. The EEPROM is ready to begin operation as soon as V_{DD} rises to be within operating limits (a power up time of zero).
2. The amount of data written fits into one page and writing takes place using the block write capability.
3. The write time of the EEPROM is only that required to perform the write operation of the EEPROM and I ignored any processing and communication time of the MCU and SPI interface. (This assumption is the opposite of that used for MRAM, which only requires communication time because the write time is so short that it can be considered to be zero.)
4. The EEPROM is powered directly from the microcontroller I/O and uses a small (0.1µF) decoupling capacitor. (**Table 1**)

Table 1

	Number of SPI Bytes	Capacitance (uF)	Rise Time Energy [3] (µJ)	Write time (ms)	Write Energy [1] (µJ)	Total Energy (µJ)	Energy/Data Byte [2] (µJ)
EEPROM	4	0.1	0.44	5.0	49.5	49.9	12.5
	46	0.1	0.44	5.0	49.5	49.9	1.0
Notes	1. Write Energy is V_{DD} x Write Current x time. 2. The number of data bytes is 4 less than the number of SPI bytes due to the overhead required. 3. Rise Time Energy = $1/2\,CV^2$						

Energy Consumed During Write Operations for EEPROM

Energy Used by a Serial Flash

Serial flash has much higher write and standby currents. I used a standby current of 50µA, a write time of 3ms and a write current of 15mA.

Since Flash is similar to EEPROM I made three of the same assumptions, (1) zero power up time, (2) the data fits into one page and (3) the write time is so long that I ignored the communication time. I also assumed that the Flash writes are to a pre-erased page. (**Table 2**)

Table 2

	Number of SPI Bytes	Capacitance (uF)	Rise Time Energy [3] (µJ)	Write time (ms)	Write Energy [1] (µJ)	Total Energy (µJ)	Energy/Data Byte [2] (µJ)
Flash	4	1.0	4.41	3.0	148.5	152.9	38.2
	46	1.0	4.41	3.0	148.5	152.9	3.3
Notes	1. Write Energy is V_{DD} x Write Current x time. 2. The number of data bytes is 4 less than the number of SPI bytes due to the overhead required. 3. Rise Time Energy = $1/2\,CV^2$						

Energy Consumed During Write Operations for Serial Flash

Energy Used by an MRAM

For MRAM there is only Everspin Technologies who have commercially available products, so I used the MR25H256, 256Kbit serial SPI MRAM.

Table 3 shows the energy per data byte is lowest when all of the energy from the decoupling capacitor is used. The decoupling capacitor size should be chosen to match the amount of data that is typically acquired by the system.

Table 3

Number of SPI Bytes	Capacitance (µF)	Rise Time Energy [4] (µJ)	t_{PU}	t_{PU} Energy [1] (µJ)	Write Time @ 40 MHz (µs)	Write Energy [2] (µJ)	Total Energy (µJ)	Energy/Data Byte [3] (µJ)
4	1.0	4.41	400	0.14	1.6	0.06	4.6	1.15
46	1.0	4.41	400	0.14	10	0.48	5.0	0.11
Notes	1. The energy consumed by the MRAM during the power up delay (t_{PU}) is given by V_{DD} x Standby Current x Time. 2. The Write Energy is given by V_{DD} x Active Current x Time. 3. The number of data bytes is 4 less than the number of SPI bytes due to the overhead required. 4. Rise Time Energy = $1/2\,CV^2$							

Serial MRAM Energy Consumed During Write Operations

A 1µF capacitor allows the writing of 50 bytes (46 data bytes) on the SPI bus at 40MHz with the MRAM consuming 27mA. This calculation is the source of using 46 bytes for comparisons.

Energy used by MCU and LDO

For the MCU, I assumed that it takes 100µs to wake up, make a measurement, and communicate the result to non-volatile memory and any housekeeping required. During this time, I assumed an active current consumption of 500µA (typical of small microcontrollers running at ~5MHz). This gives us an energy consumption of 3.3V x 500µA x 100µs = 0.165µJ per data acquisition. In addition to the energy to make an acquisition, I added the energy required to keep the microcontroller active during the non-volatile memory write.

When not acquiring or storing data, the MCU was in a sleep state consuming 5µA.

The power supply was assumed to be an LDO that consumes 1µA during all phases of operation, active and sleep.

Results Summary

Table 4

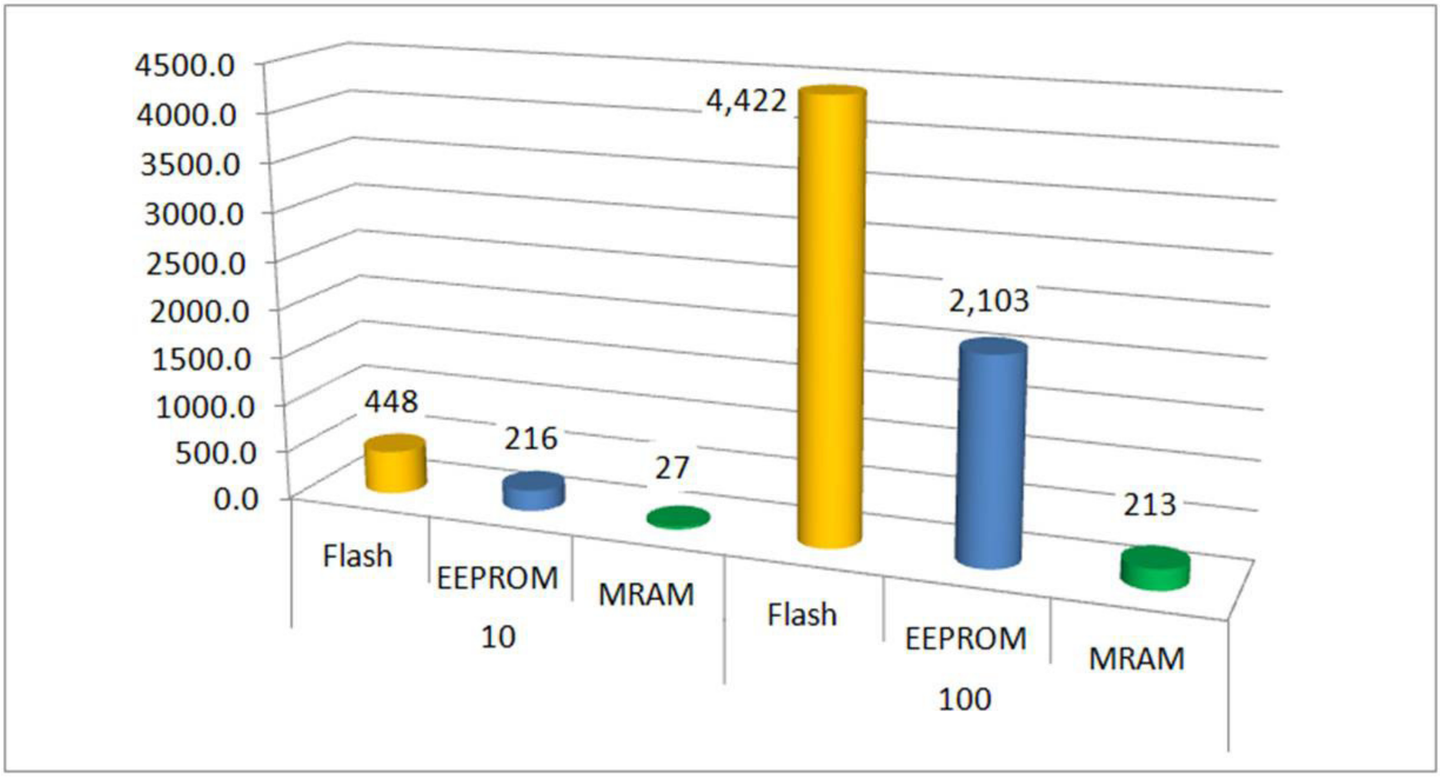
Memory Type	Number of Acquisitions/sec [4]	Active Time [1] (ms)	Sleep Time [2] (sec)	Energy Consumed Sleep Time [3] (µJ)	Energy Consumed Making Acquisitions (µJ)	Total Energy (µJ)
MRAM	10	4.5	0.995	6.6	20.6	27.2
	100	4.5	0.955	6.3	206.4	212.6
EEPROM	10	5.05	0.950	6.3	93.4	99.6
	100	5.05	0.495	3.3	933.7	937.0
Serial Flash	10	35.0	0.965	6.4	441.8	448.1
	100	35.0	0.650	4.3	4,418.0	4,422.3
Notes	1. The Active Time is given as 100µs of MCU time + Storage Time. 2. The Sleep Time is given as 1 - Active Time. 3. Energy consumed while asleep is given by 3.3V x 2µA x Time. 4. Four bytes per acquisition.					

Summary of the Energy Consumed per Acquisition

Graphic Summary

The results are graphically summarized in **Figure 3**.

Figure 3



Total Energy by Type and Number of Acquisitions (μJ)

Conclusions

The write time of a non-volatile memory significantly affects the total energy consumption of a system. For systems with a low duty cycle the effects are less pronounced and become more pronounced as the rate of acquisitions increase.

The write time of EEPROM and Flash significantly increase the energy consumption of the MCU because they cause the MCU to be active for longer. Energy consumption could be reduced if the MCU was in a sleep mode while the writes to EEPROM and Flash complete. However, the energy consumed by the EEPROM or Flash is the majority of the energy consumed by the system so having the MCU in sleep mode will not effect the overall consumption significantly.

It is clear that the lowest energy consumption can be achieved with a fast-Write, non-volatile memory that is power gated.

Why Power Gating has more of an effect on energy consumption than Sleep Mode

EEPROM has low standby power consumption so operating it with V_{DD} always present could be considered. However, the write energy of just one Write operation to EEPROM is equivalent to it being in standby mode for about 15 seconds. Again, the write energy is dominant.

Power gating really applies to MRAM where the fast Write time significantly reduces the amount of energy used. The energy consumption of an acquisition using MRAM is 1.54µJ. This is the same as a 3.3V EEPROM with a standby current consumption of 1µA being in standby for 0.46s.

The write energy is dominant over standby energy.

About the Author

Duncan Bennett is a Product Marketing Manager on the marketing staff at Everspin Technologies, Inc. He is stationed in London. You can reach Duncan at: **duncan.bennett@everspin.com.**

This article first appeared in eeTimes *DesignLines* (Memory) on December 9, 2014. **http://www.eetimes.com/author.asp?section_id=36&doc_id=1324917**

Revision	Date	Description of Change
1.0	March, 2015	Initial Release.

How To Contact US

Home Page:

www.everspin.com

World Headquarters

1347 N. Alma School Road, Suite 220

Chandler, Arizona 85224

1-877-347-MRAM (6726)

1-480-347-1111

Orders and Ordering - orders@everspin.com

Sales - sales@everspin.com

Authorized Representatives and Distributors

United States, Canada and Mexico - Here

Email **-** support@everspin.com

South America - Here

Email **-** support@everspin.com

Europe, Middle East and Africa - Here

Email **-** support.europe@everspin.com

Japan - Here

Email **-** support.japan@everspin.com

Asia Pacific - Here

Email **-** support.asia@everspin.com

File Name:

Using MRAM to Optomize System Energy Consumption 021315