EVERSPIN TECHNOLOGIES

Whitepaper

MRAM Improvements to Automotive Non- Volatile Memory Storage

August 5, 2016

Author(s):

Chinh Nguyen (Ford)

Dona Burkard (Ford)

Kelvin Dobbins (Ford)

Chuck Bohac (Everspin)

Abstract

Automotive powertrain modules use flash memory technology to retain critical control and diagnostic information during power off (keep-alive memory (KAM) and non-volatile memory (NVM)). Complex software must be designed to maximize the lifecycle of these devices because they have a limited number of write cycles.

MRAM (Magneto resistive Random Access Memory) has the potential to eliminate this complexity and make the process of managing KAM and NVM easier and more robust. This paper demonstrates using off-board MRAM devices with a next generation of powertrain microprocessor. The prototype boards integrating the latest powertrain microcontroller, with the Everspin MRAM MRA16A (2 pieces of x16 bits) and MR2xH50 (@ a SCK 40MHz) chips were created.

An investigation was performed evaluating the MRAM capabilities for storing and retrieving data during simulated key-off and key-on events.

Introduction

The automotive industry is being introduced to new non-volatile memory such as MRAM, Phase-control Memory (PCRAM) and RRAM. These new memories offer improvements in storing and retrieving data during automotive change of state conditions. Powertrain Controls Research and Advance (PCRA) team worked with Everspin and a micro-controller supplier to rapid prototype external MRAM with the next generation powertrain controller.

Non-volatile memory

Currently, Engine Control Units (ECU's) use non-volatile memory (NOR FLASH) to store logging data (DFLASH). The data is stored in the DFLASH and logged after the ignition key is turned off (otherwise known as KO (Key Off)). This data is used to refine coefficients of the algorithm used to control the drive train as well as to determine if sensor input has exceeded predetermined limits. Non-volatile data must be stored (such as individual injector trim data captured from barcode reads during engine and vehicle assembly, transmission solenoid characterization data, and permanent P-Codes, among others). KAM-type data, such as adaptive tables, needs to be retained after a key-cycle, but can be relearned if lost.

Limitation of DFLASH and current issue

The bank swap operation is the limitation of current DFLASH. It takes a several seconds for writing non-volatile data to DFLASH. When a bank swap occurs, the unused bank of the D- Flash is erased, the file system is established on the erased bank, and finally the old bank is searched to find the latest data which is copied to the new bank. There is plenty of time for the key to come back on during a bank swap which is where most problems occur. The number and sizes of NVRAM data structures supported in this manner can differ between different application versions and vehicle types. The frequency of update requests is variable, some being based upon the occurrence of specific events (e.g., Vehicle configuration info changed or new diagnostic events need to be stored).

Potential usage of MRAM in non-volatile memory

As opposed to electron charge to store data that can leak and breakdown with usage, time and temperature, MRAM (Magnetic Random Access Memory) uses a 1 transistor – 1 magnetic tunnel junction (MTJ) architecture with the magnetic state of the MTJ as the data storage element. Because MRAM stores data as a magnetic state, it offers numerous and significant advantages over existing non-volatile memories. The benefits of MRAM over other non-volatile solutions are:

- FLASH like non-volatility
- No Program/Erase cycles (fastest WRITE cycles)
- Fast, SRAM interface
- Symmetric Read/write cycles
- Byte Addressability
- 35ns/2Bytes with parallel (x16) interface
- 0.4us/2bytes with serial interface (using SPI interface operating at 40MHz)
- Unlimited endurance (unlimited usage)
- Long data retention (20+ years at high temperatures)

MRAM with powertrain microcontroller demonstration

One of the challenges facing the team (and industry to date), is that these new memory technologies (such as MRAM (which is already embedded on some processors today)) have not yet been embedded on powertrain silicon. Therefore, for this demonstration, the team employed a discrete, "off-processor" solution. In this demonstration, the team designed MRAM evaluation boards to interface to the Powertrain micro-controller mother board. New low level drivers (LLDs) were developed to replace the LLD's provided by the powertrain silicon supplier. In this demonstration, the newly revised LLD's were utilized to verify the serial/parallel MRAM protocols, read/write operations in different modes, and timing cycle requirements (by running the applications on the prototype boards with various clock frequencies).

Hardware implementation

In this demonstration, the latest powertrain microcontroller utilized both serial and parallel MRAM interfaces. Because of the complicated protocols and memory controllers, the demonstration required verifying all operation modes and the entire address range of MRAM in both parallel and serial communications (Fig 2.2)

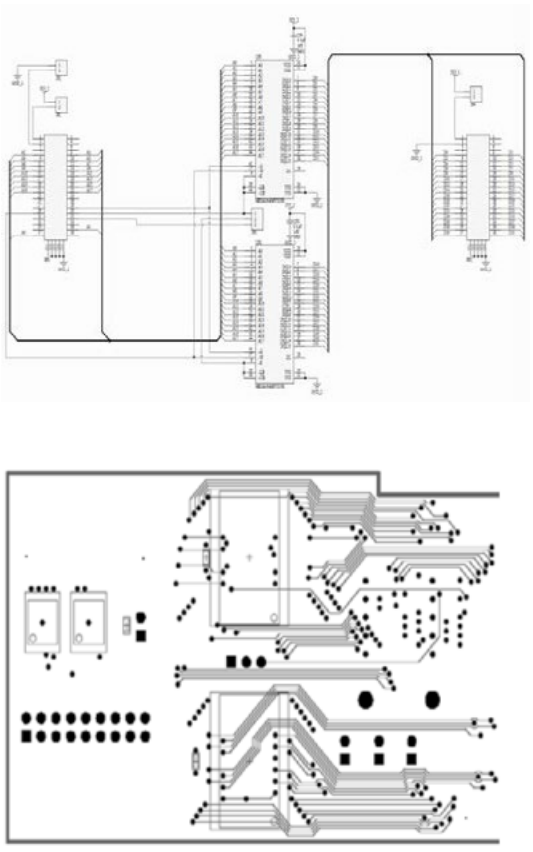

Figure 2.1 Parallel MRAM schematic and layout

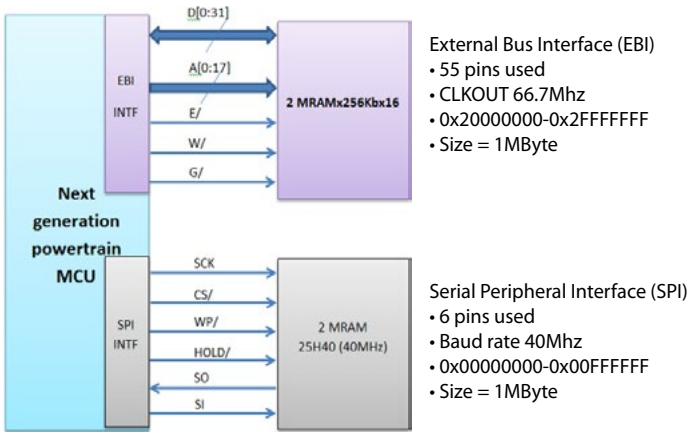

Figure 2.2 Powertrain MCU and MRAM interfaces

In this demonstration, two parallel MRAM (2 x MR2A16A) and two serial MRAM (2 x MR2xH50) devices were used. The LLDs were designed to match the EBI and SPI buses of the powertrain processor, respectively. For the parallel devices, the accessing data bus was flexible with 32bits/16bits with an internal clock of 66.667MHz while the serial devices used a 40MHz clock.

Software implementation

The team developed the software required to implement the MRAM technology with a complex powertrain application. This included developing low-level drivers and other software layers. These LLDs replaced the current suppliers' LLDs. These drivers drove the NVRAM/KAM data to MRAMs chips instead of current DFLASH following a request from the arbitrator to the LLDs (Fig 3.1).

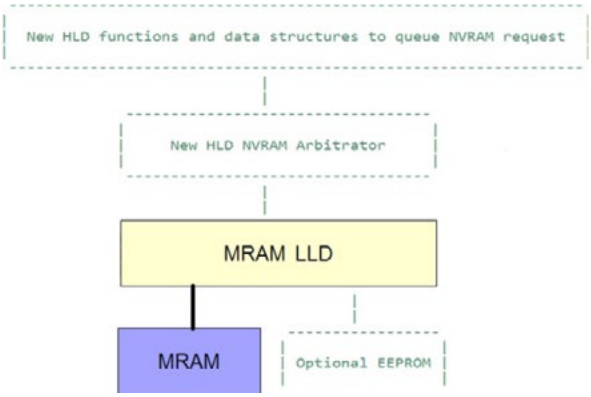

Figure 3.1 Low Level Drivers Flowchart

Serial and parallel MRAM timing analysis

This demonstration was run on the powertrain microcontroller board and debugger tool was used to capture the waveform and measure timing.

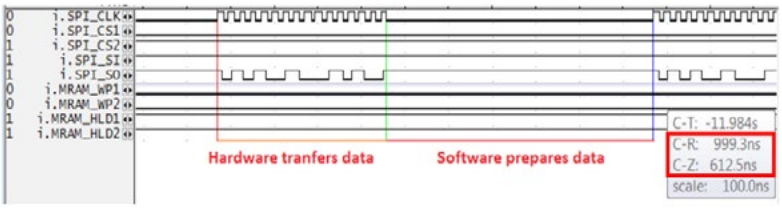

Figure 4.1 16-bit Data output using SPI MRAM

Figure 4.1 shows the waveform for transferring 16 bit data to the output pin using the SPI protocol. A 40MHz clock was used in this transmission. The total duration of the writing cycle is around 1.0ms which includes both hardware and software delays. The hardware delay is the period for SPI to transfer 16 bit data from hardware buffer (387.0 ns). The software delay is the period for the software to clear flags and wait for new data (612.5 ns).

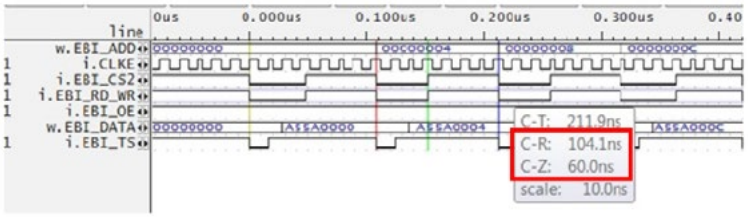

Figure 4.2 32-bit data output using EBI MRAM

Figure 4.2 shows the waveform for transferring 32 bit data to the output pins using the x32 parallel interface. The total duration of the writing cycle is around 104.1ns which includes the hardware and software delays. The hardware delay is around 44.1ns to transfer data from buffer to output pin while the software delay is around 60.0ns to prepare the new data.

	Operations	32/16bits	20KBytes
EBI Bus	Read (32 bits)	103.2ns	1.05ms
	Write (32 bits)	103.3ns	512.1µs
SPI Bus	Read (16 bits)	1.46µs	14.95ms
	Write (16 bits)	1.00µs	10.24ms

Note - Significant throughput improvement with DMA

Figure 4.3 Read and Write cycles between SPI and EBI MRAM

Figure 4.3 shows that the read/write cycles of EBI chip are much faster than SPI chip. The EBI devices also support flexible 16bit/32bit access data. However, the SPI devices support the memory protection mode which allows users to protect data. The throughput will be improved if SPI MRAM is combined with DMA controller. The read/write cycles measured on the prototype boards are similar with timing on the MRAM specifications. The read/write operations of parallel chips is around 0.1us while SPI devices operate around 1.0us for each read/write cycle

Demonstration Software Analysis

MRAM low level drivers were integrated into powertrain application with operation system and scheduling. The read and write cycles were measured by the system clock (300MHz). Figure 5.1&5.2 show the read/write time of each partition with the different non-volatile memory interfaces for the powertrain application. The tables show that most of read/write cycles are less than 2ms. Not surprisingly, the table confirms that the 35ns parallel interface transfers data at a much faster rate than the 40 MHz serial interfaced MRAM.

With SPI MRAM, because of the hardware delay of the micro-controller (buffer receive/transmit, set/clear flag, read/write memory) and synchronization between the MRAM and micro-controller buses, the read cycles took longer time than write cycles, similarly with the parallel MRAM the write cycles took longer

than read cycles. The values displayed in 5.1&5.2 included the hardware transceiver, hardware delay (transceiver buffers, read/write memories), LLD software delay and synchronization between the MRAM and the powertrain microcontroller.

Partition	Size (bytes)	Read (ms)	Write (ms)
0	140	0.11240	0.04418
1	512	0.38762	0.13922
2	128	0.10352	0.04106
3	256	0.19830	0.07370
4	40	0.03936	0.01888
5	0	0.01338	0.00950
6	4460	3.30970	1.14532

Figure 5.1 SPI MRAM; CLK 40MHz

Partition	Size (bytes)	Read (ms)	Write (ms)
0	140	0.01330	0.01546
1	512	0.03450	0.03906
2	128	0.01204	0.01322
3	256	0.01982	0.02166
4	40	0.00602	0.00768
5	0	0.00502	0.00592
6	4460	0.27206	0.30540

Figure 5.2 EBI MRAM; CLK 66.666MHz

Features	EBI MRAM	SPI MRAM
Read/write cycle	Faster (in ns)	Slower (in µs)
Hardware resource	55 pins	6 pins
Extendable	3 chips	7 chips
Data bus	16/32 chips	1 bit input, 1 bit output
Capacity	.05MB/chip	0.5MB/chip
Debugging	Easier	More difficult
Reliability	Good	Good
Hardware inf/config	Simple	Simple
Hardware protection	No	Yes

Conclusion

In this demonstration, we verified the different powertrain operation modes with both EBI and SPI interfaced devices. Various types of data were read and written in the entire range of address space. In general, the MRAM operations and timing are similar to 32 bit microcontroller specifications and timing. Moreover, the performance and throughput of MRAM devices are acceptable for today's non-volatile memory compared with DLFASH.

Future automotive powertrain controllers may have a need for faster and more robust non- volatile and keep-alive memory when compared with today's DFLASH. As demonstrated in this project, the use of MRAM for non-volatile memory storage significantly improves the performance of write data transfers during time-critical states. The increased memory storage available with MRAM devices may also enable the storage of more diagnostic data.

In summary:

- MRAM has direct write cycles therefore eliminates the need for complex software routines when compared to today's DFLASH.
- MRAM has larger memory storage when compared to the DFLASH usage (a hundreds of Kilobytes) in current automotive microcontrollers.

The future of MRAM technology and commercial markets

Because standard CMOS wafers are used as the base wafer for MRAM technology, integrating MRAM technology with microprocessor and System-on-a-Chip (SOC) designs is realizable. eMRAM (embedded MRAM) will offer significant advantages in both performance and time-to- market for such designs.

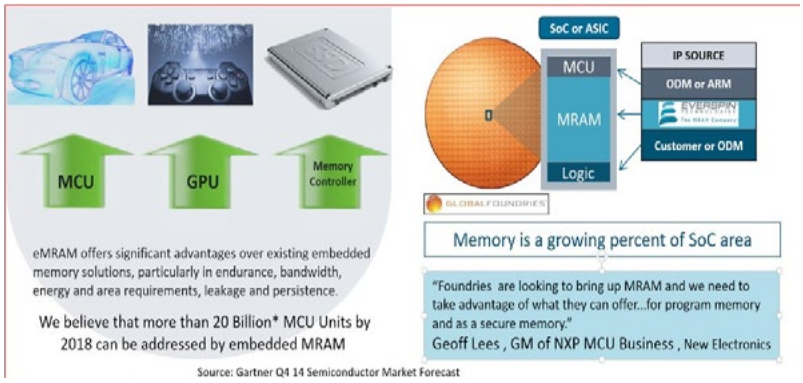

Fig 7.1 MRAM in commercial market

Need	eFLASH	eSRAM	eDRAM	MRAM
Low Cost	More steps	Large 6T cell	More steps	Yes
Byte Addressable	No	Yes	Yes	Yes
Low Latency	No	Yes	Yes	Yes
Persistent	Yes	No	No	Yes
CMOS Compatible	High Voltage	Yes	No	Yes
Scalable	No	High Leakage	Yes	Yes
Low Energy	No	No	No	Yes

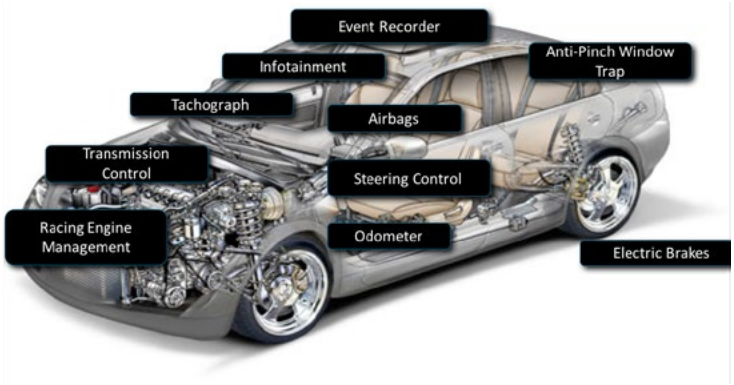

Fig 8.1 MRAM applications in automotivel markets

MRAM Applications in the Automotive Market

There are numerous applications for MRAM technology in the automotive market. Because MRAM technology integrates with standard CMOS technology it can be embedded on powertrain microprocessors (MRAM is embedded on some non-automotive microprocessors today). However, at this time the most economical approach to integrating MRAM into Powertrain silicon is with die stacking for the following reasons:

- Removing the memory array from powertrain silicon reduces its die size which increases the number of die per wafer while increasing the die yield. Both of these actions lower the cost of powertrain silicon
- Advancements in die stacking technology have matured to acceptable quality levels. Therefore, stacking a Known Good Die (KGD) of MRAM on a Known Good powertrain die offers the least expensive solution as it integrates the lowest cost CMOS die with the lowest cost MRAM die.

Until integration of next generation ST-MRAM offers the price/performance ratio required by the automotive market (estimated to be approximated 5+ years), the well proven, currently available, robust MRAM technology of today appears to be the most optimal way to integrate MRAM technology into automotive applications.

Contact Information:

Author:

Chuck Bohac

Director, Application Engineering

How to Reach Us:

www.everspin.com

E-Mail:

support@everspin.com

orders@everspin.com

sales@everspin.com

USA/Canada/South and Central America

Everspin Technologies

5670 W. Chandler Road, Suite 100

Chandler, Arizona 85226

+1-877-347-MRAM (6726)

+1-480-347-1111

Europe, Middle East and Africa

support.europe@everspin.com

Japan

support.japan@everspin.com

Asia Pacific

support.asia@everspin.com

Everspin Technologies, Inc.