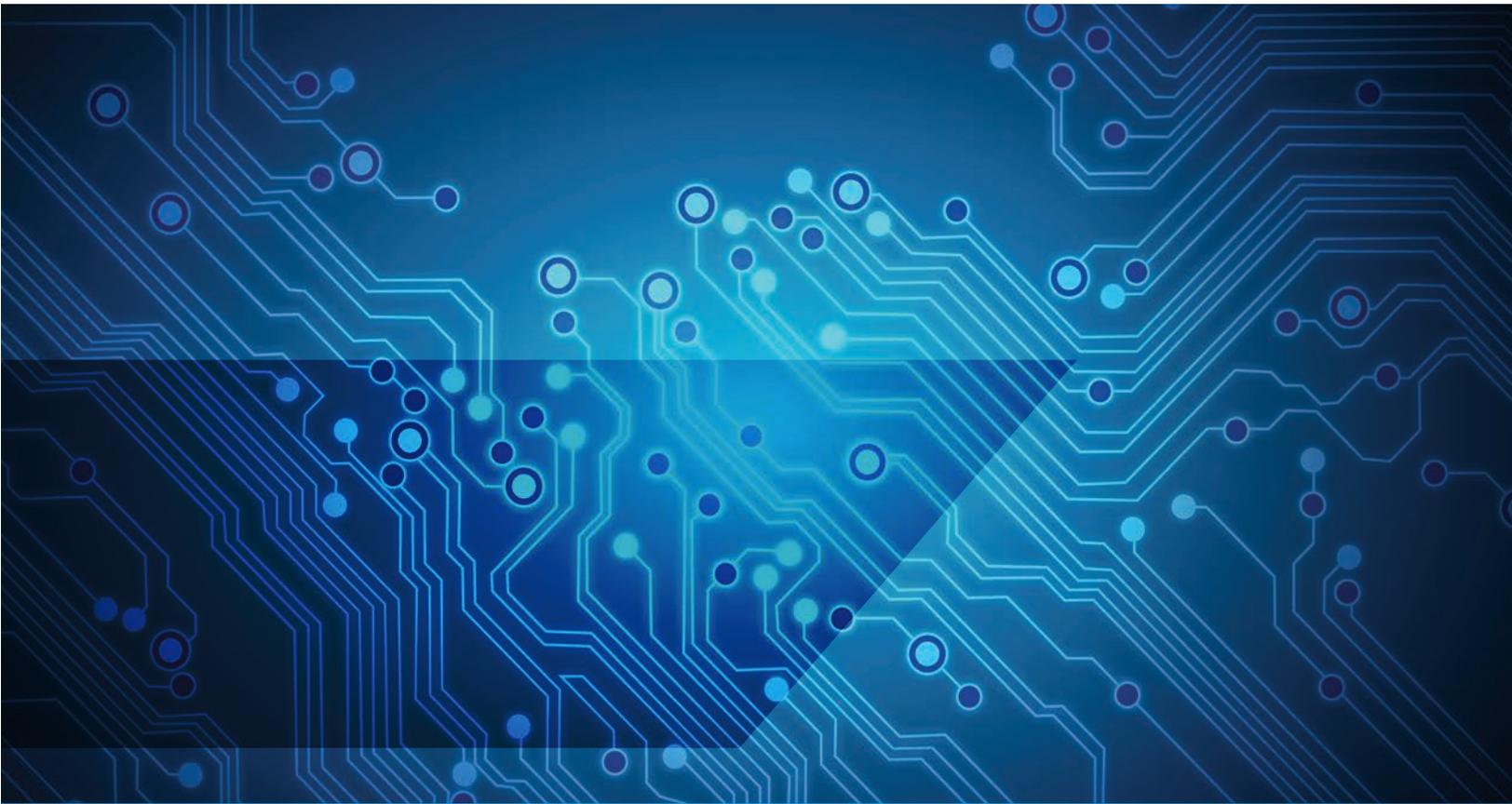


ADVANCED MRAM TECHNOLOGY

USING MRAM IN PLACE OF SRAM



USING MRAM IN PLACE OF SRAM

LOWER COST, LOWER ENERGY, NON-VOLATILE, AND RAD HARD

Systems-on-chip (SoCs) rely heavily on SRAM technology for high-speed access to frequently used data. SRAM is ubiquitous: probably every SoC, ASSP and processor uses SRAM, and, for a number of devices — processors in particular — the majority of the total die area may be occupied by SRAM.

Performance is SRAM's primary benefit, and it comes at the expense of both die area and power. An SRAM memory is very large compared to most other types of memory: for the same number of bits, an SRAM is 20-30 times bigger than a DRAM and probably more than 100 times bigger than a flash memory. SRAM's speed, flexibility and easy integration to CMOS processes comes with a significant die cost penalty.

SRAM's other weakness is leakage, which leads to standby power dissipation. For battery-powered Internet of Things (IoT) devices, high standby power can be a deal-breaker, since many of these devices will spend the vast bulk of their time idle. If the battery drains during idle times, battery life becomes severely compromised. With that in mind, SRAM's performance comes at a high cost.

In addition, embedded systems that use SRAM must store any persistent data when the system goes to sleep. Storage is also needed for code and for data used to configure or personalize the system on power-up. Flash non-volatile memory (NVM) has traditionally served this role, but flash technology has significant writing, reading and erasure limitations that add complexity to the system.

For several years, a "More than Moore" industry effort has endeavored to find a memory technology that could replace SRAM and eliminate the need for NVM; yet, until recently, no suitable replacement memory technologies have been found. Numerous candidates have come and gone, plagued by expense, complexity or any number of other issues that rendered them unfit for commercial production. However, one such replacement technology is poised to gain ground on SRAM: MRAM.

MRAM, or magnetoresistive RAM, is an emerging persistent memory technology that has achieved commercial production. It has major benefits as compared to SRAM and traditional NVM:

- It's byte-addressable, like SRAM, and unlike NVM.
- Performance is comparable to that of SRAM and much higher than that of NVM.
- MRAM can be built to match the endurance of SRAM, which is many orders of magnitude higher than that of NVM.
- MRAM has none of the complex downsides associated with NVM, like sector erase and wear-leveling.

While these features align MRAM more closely with SRAM, there are four key characteristics that differentiate MRAM from SRAM: cost, leakage current, non-volatility and radiation hardness. When it comes to these four considerations, MRAM captures the best characteristics of both SRAM and NVM technologies, while inheriting none of the downsides.

LESS THAN HALF THE COST OF SRAM

	SRAM	MRAM
SPEED	FAST	FAST
BITCELL	LARGE	SMALL
LEAKAGE	HIGH	NONE
VOLATILE	YES	NO
RAD HARDNESS	NO	YES

FIGURE 1.
MRAM COMPETES WITH SRAM ON SPEED AND BEATS SRAM ON ALL OTHER CRITICAL PARAMETERS.

AN MRAM BLOCK WILL BE ABOUT ONE-THIRD THE SIZE OF AN EQUIVALENT SRAM BLOCK.

Fabless companies typically rely on their foundries or specialized memory providers for their SRAM blocks, and foundries tightly optimize their SRAM bitcells, holding them as critical intellectual property. It's not uncommon for a technology node to be selected not on the basis of required overall performance, but specifically because of the SRAM technology available at that node. If the SRAM requirements dictate a more aggressive node than the rest of the circuit requires (especially circuits that do not scale across nodes, such as analog or high-voltage circuits), then the chip costs can rise substantially. Display-driver ICs are a prime example of this phenomenon.

The primary problem with SRAM is the size of the bitcell — even if it is highly optimized by the foundry. The SRAM bitcell uses six to eight transistors. There are even so-called “non-volatile SRAM” cells that require 12 transistors, which places them out of cost range for any applications that don't have a driving need for persistence at any cost.

Unfortunately, SRAM sizing is getting worse, not better, as technology nodes advance. The most common figure of merit for memory bitcell size is “F-squared” — the bitcell size relative to the size of the technology node. In a 55-nm node, for example, $F=55 \text{ nm}$. For years, SRAM bitcells were $\sim 180F^2$ — that is, they were 180 times the square of the process node. However, in the most advanced process nodes, bitcells are becoming several hundred F-squared. For example, in a 2017 IEDM conference paper, GLOBALFOUNDRIES announced that its SRAM bitcell in the 7-nm node will be $550F^2$, while Intel is announcing that its 10-nm node bitcells are greater than $400F^2$.

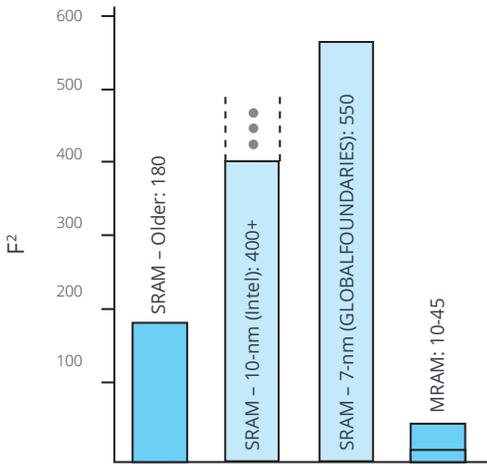


FIGURE 2.
MRAM HAS A MUCH SMALLER CELL THAN SRAM — AND THE ADVANTAGE WILL INCREASE WITH ADVANCED PROCESS

MRAM, by contrast, uses a single transistor in its memory cell. That transistor is combined with the magnetoresistive structure that provides the storage, so no other support transistors are needed within the memory array. As a result, a full MRAM memory block, including peripheral circuits, will be about one-third the size, or two-thirds smaller than an equivalent SRAM block. We believe that this relationship will become more pronounced at more advanced nodes — that is, MRAM may be one-quarter the size of SRAM, and perhaps even less, at the 10-nm and smaller nodes.

The key to this characteristic is the magnetoresistive element, referred to as the magnetic tunnel junction (or MTJ). It requires three additional processing steps as compared to SRAM (which is pure CMOS). This sets up a subtle cost dynamic: these extra steps add 5-11% to the overall wafer cost, but the die size savings are so substantial, especially for designs making heavy use of SRAM, that replacing it with SRAM with MRAM results in a significant reduction in die cost.

NO LEAKAGE CURRENT AND NON-VOLATILITY

MRAM ARRAYS DO NOT LEAK. THEY CAN BE POWERED DOWN WITHOUT LOSING THEIR CONTENTS.

One of the primary challenges of SRAM is leakage current — anywhere from 5 to 100 pA per bit at room temperature. That current can rise by an order of magnitude with hot temperatures. Each leaking memory cell contributes to battery drain, a particularly important consideration for the enormous number of new battery-powered devices being created for the Internet of Things (IoT).

SRAM technologists must walk a fine line between achieving acceptable SRAM performance and power. Unlike other parts of a circuit, an SRAM array cannot be completely powered down when idle without losing its contents. As a compromise, designers look for a “retention” voltage that’s not high enough to operate the circuit, but is high enough to maintain the SRAM contents. This lowers leakage during idle periods, but does not eliminate it.



MRAM bits, by contrast, do not leak; the only time there is a current path is when the bit is being written or read. As with SRAM, the MRAM peripheral circuits — which are standard CMOS — do contribute to some leakage current. But in an SRAM, the array leakage swamps the peripheral leakage.

With MRAM, there is no array leakage when idle. The amount of actual current saved by using MRAM depends on the size of the array; the following table illustrates the possible savings. (Note: SRAM leakage current varies widely among foundries, processes and process targets — for example, “ULP” processes are designed for low power and leakage, while high-performance SRAMs will have much higher leakage currents.)

BLOCK SIZE (BITS)	4M	4M	4M	4M	64M	64M	64M	64M
BITCELL CURRENT (PA)	5	100	50	1000	5	100	50	1000
CONDITIONS	ROOM	ROOM	HOT	HOT	ROOM	ROOM	HOT	HOT
SRAM BITCELL LEAKAGE	21 MA	420 MA	210 MA	4.2 MA	340 MA	6.8 MA	3.4 MA	68 MA

These enormous leakage numbers can be devastating to battery life, not to mention related considerations like packaging costs (since a more expensive package is required to handle the heat) and system cooling requirements.

Furthermore, MRAM's non-volatile characteristics form a powerful argument for an aggressive power-gating strategy. Typical SRAM power gating requires some, or all, of the SRAM contents to be stored to non-volatile memory somewhere. However, because MRAM is persistent, you can power it down with no loss of contents and no need to save the contents to some other memory first.

IMMUNE TO SOFT ERRORS AND RAD HARD

MRAM CONTENTS ARE ROBUST AGAINST RADIATION.

SRAM bits are susceptible to having their state change as a result of radiation or being struck by a particle. Worryingly, the smaller an SRAM cell is, the more likely it is that a single-event upset (SEU) will affect more than one bit. This is a significant concern in applications where data integrity is paramount, including data centers and the safety-critical automotive, military and aerospace markets.

MRAM bits do not lose their contents during an SEU. They can operate with immunity in harsh radiation environments.

APPLICATION EXAMPLE: DISPLAY DRIVER INTERFACE

Display drivers are a good example of an application that uses a large amount of SRAM. The SRAM array typically makes up about 40% of the total die area. Replacing the SRAM with MRAM lowers the die size by about 27%. Factoring in the MRAM wafer cost adder of 5%-11% produces an overall die cost savings of 25%.

This is one of those applications where the technology node is driven by the SRAM needs, not the rest of the circuitry. A given die may need 55-nm technology for the SRAM, but with MRAM you can achieve an equivalent die cost on a 110-nm node despite CMOS circuits that consume more than five times the area of a 55-nm version.

With the prospect of relative SRAM bitcell size increasing on more aggressive process nodes, MRAM's advantage over SRAM in this and similar applications is likely to grow substantially.

APPLICATION EXAMPLE: BATTERY-POWERED IOT DEVICE

IoT device designers using SRAM have some tough decisions to make in order to reduce power. One possibility is to rely on an array retention voltage during "sleep" rather than fully powering down the array. But if a complete power-down of the SRAM is desired, then you must save data to flash memory before shutting off the power. This technique requires extra time, both when going to sleep (so that data can be stored) and when waking up (so that data and code can be restored).

The steps necessary to fully go to sleep with SRAM consume additional energy, and, therefore, each sleep/wake cycle further drains the battery. A significant contributor to both complexity and energy consumed when using flash memory is the fact that variables can be written to flash incrementally, but flash erasure affects an entire sector. As a sector fills, at some point it, or another sector, must be erased to allow further writes.

In addition, when a variable is updated, its existing location in flash can't be rewritten; it must be invalidated, with a new entry written elsewhere. Because of multiple, or, possibly, out-of-date versions of the same variable, extra metadata must be stored.

These flash characteristics contribute significantly to both the overhead time spent and the overhead energy consumed when executing wake/sleep cycles. In fact, a key system design decision is how long the SRAM must be "asleep" before the energy savings of the powered-off state more than compensate for the overhead of the sleep/wake sequence. The following table illustrates one possible scenario, depending on how many variables must be managed.

# DYNAMIC VARIABLES	5	10	100
# STATIC VARIABLES	10	10	10
CODE SIZE (BYTES)	50,000	50,000	50,000
# WAKE CYCLES/HOUR	1	1	1
OVERHEAD TIME PER DAY (S)	0.1	0.2	1.5
OVERHEAD ENERGY PER DAY (J)	0.07	1.4	1.4

- “Dynamic” variables are those that might change during execution; they must be saved before going to sleep.
- “Static” variables are fixed values; they and the code must be loaded at wake-up. Neither static variables nor code needs to be saved when going to sleep.
- IoT devices typically wake up periodically to measure something, report the measurement, and then go back to sleep.
- The “# wake cycles/hour” line shows how often this happens
- The final line shows the resulting extra energy consumed each day due to the overhead of saving to and restoring from flash memory.

SUMMARY

SRAM's high cost and high energy consumption have created demand for a replacement memory technology. Using MRAM in place of SRAM is a “More than Moore” solution, with dramatic improvements in both performance and efficiency.

This is possible because MRAM:

- Competes on performance with SRAM
- Has an SRAM-like interface: it's word- and byte-addressable
- Has a much smaller bitcell than SRAM, yielding significant die size and cost savings
- Is persistent, and therefore needs no NVM for backing data up when powering down
- Has much lower energy consumption when sleeping, with no bitcell leakage and without the need to consume energy when saving and restoring data in order to sleep
- Is radiation-hard, naturally surviving in harsh environments

SoCs can be made with lower power, equivalent performance and higher reliability by turning to MRAM for applications that formerly used SRAM. This will also significantly simplify or even eliminate the need for any NVM. The result will be a cheaper, simpler, more competitive system well-suited for low-cost, battery-powered devices.

[FOR MORE INFORMATION ABOUT HOW MRAM CAN SIMPLIFY YOUR SoC DESIGN AND SAVE POWER, CONTACT SPIN MEMORY](#)



45500 Northport Loop West
Fremont, California 94538
(510) 933-8200 main
(510) 933-8201 fax
Email: info@spinmemory.com

www.spinmemory.com