Making MRAM Work in Today's Applications

Brent Yardley (IBM)
yardleyb@us.ibm.com
Chief Hardware Engineer, Master Inventor, IBM System

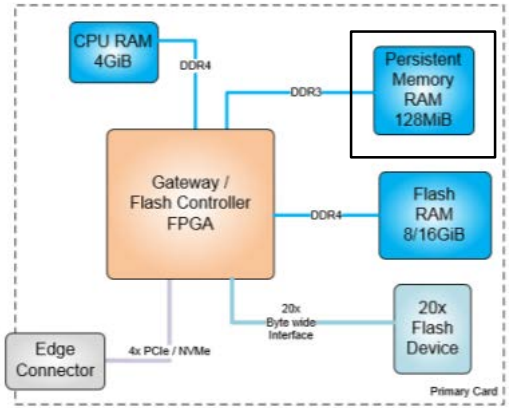
Abstract:

MRAM is the fastest persistent memory media available on the market today, making it uniquely capable of satisfying both high performance and non-volatility requirements for today's demanding applications. The difference between "satisfying requirements" and "exceeding expectations" is determined by the development cycle "details" of MRAM implementations. This presentation covers implementation details of shipping MRAM solutions regarding performance, reliability and lifetime for today's users, and considers how coming MRAM improvements will affect these implementations.

Agenda

- Current MRAM Use Case
- What is required to implement MRAM in current & future designs
- Looking to the future of MRAM implementations

FlashCore™ System Overview:



MRAM Use Case in FlashCore™ Module :

- Write Data Buffer/Cache
- Read Modify Write Buffer
- Flash Status Back Up on Power Loss
- Journal Buffer
- Flash Firmware Tables Storage
- NVMe Persistent Data (includes state dumps)

IBM's FlashCore™ Module (FCM) with STT-MRAM

- With the help of 256 Mb STT-MRAM low latency data persistence is achieved in IBM's Flash Core Module
- Significant real estate saving achieved through elimination of super-caps, and reduced footprint of Tantalum caps

The Path to Realization

- Power de-staging electrical noise
- Endurance Lifetime Tracking
  - Soft vs Hard Errors
- Temperature Induced Errors & Data Retention
  - data decay & using destructive reads
- BER Management for DRAM Equivalent Reliability
  - Stronger System ECC – DEC/TED
- Data Management & Wear Leveling

FlashCore™ Module 1.0 MRAM Implementation

- Configuration
  - 4+1 x8 configuration using 256Mb device for a total of 128MiB of capacity
- System level ECC Engine
  - 80 bit reads, two 32 bit data words, 16 bit codeword, t=2 correction, w/ TED
  - Goal is to approach DRAM like BER
- Modification of DDR3 memory control to support SCRAM feature
  - Also added asserting reset after SCRAM complete
- Added read scrubber for data retention
  - Read scrubbing was a safe way to scrub the full memory area without the need to lock
- Access counters for wear statistics
  - Used to provide statistical data on how MRAM is is accessed in the system
- Wear leveling
  - Used address walking method to make sure cache areas wear evenly

FlashCore™ Module Future Implementation

- Configuration
  - 2+1 x16 configuration using 1Gb DDR4 devices
    - Doubling of the capacity to 256MiB
- System level ECC Engine
  - Take advantage of 48 bit interface
  - Single 48 bit read, 32 bit dataword, 15 bit codeword, with t=2 w/ TED
  - Remove straddling two address of data, increases data protection
- Modification of DDR4 memory control to support SCRAM
  - Continue use of asserting reset after SCRAM complete
- Implement Start-Gap to wear level across the whole capacity
  - Read Scrubber for data retention more effective in combination with Start-Gap allowing a rotation through all of memory, while reading a single address, and implement of gap progression features
- Data scrambling using address bits seeding
  - Allows for uniformity in data writes across all data sets
- Implementation of MFG screens similar to NAND

Future Considerations

- Density
  - Okay for current implementation needs (Write Cache / RMW Buffer / PMR)
  - Higher capacity usage limited by current $/GB
- Cost
  - One of the largest contributing factors to not being able to implement additional capacity and true DRAM replacement
  - BER drives reservation of non-user capacity (data protection bits/ECC)
- Standardized interfaces
  - Persistent Memory needs a unified standard (PM-DDR)
    - Implementers and architects struggle with the not having commonality
    - Would like to see common feature set, with room to have vendor uniqueness
- RBER Reductions
  - Need to get to a point where t=2 is not required
  - DRAM equivalence to lower interface complexity
- Write Endurance
  - Need to increase write endurance by 1 to 2 orders of magnitude to reduce complexity of wear leveling designs
  - Non-Destructive Reads is a key enabler
- Performance
  - Align with DDR speed interfaces and widths