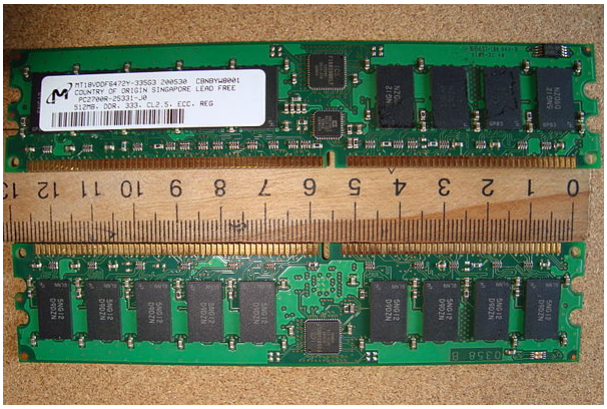
WIKIPEDIA

ECC memory

Error-correcting code memory (**ECC memory**) is a type of computer data storage that can detect and correct the most common kinds of internal data corruption. ECC memory is used in most computers where data corruption cannot be tolerated under any circumstances, such as for scientific or financial computing.

Typically, ECC memory maintains a memory system immune to single-bit errors: the data that is read from each word is always the same as the data that had been written to it, even if one of the bits actually stored has been flipped to the wrong state.[2] Most non-ECC memory cannot detect errors, although some non-ECC memory with parity support allows detection but not correction.



ECC DIMMs typically have nine memory chips on each side, one more than usually found on non-ECC DIMMs (some modules may have 5 or 18).[1]

Contents

Problem background

Electrical or magnetic interference inside a computer system can cause a single bit of dynamic random-access memory (DRAM) to spontaneously flip to the opposite state. It was initially thought that this was mainly due to alpha particles emitted by contaminants in chip packaging material, but research has shown that the majority of one-off soft errors in DRAM chips occur as a result of background radiation, chiefly neutrons from cosmic ray secondaries, which may change the contents of one or more memory cells or interfere with the circuitry used to read or write to them.[3] Hence, the error rates increase rapidly with rising altitude; for example, compared to sea level, the rate of neutron flux is 3.5 times higher at 1.5 km and 300 times higher at 10–12 km (the cruising altitude of commercial airplanes).[4] As a result, systems operating at high altitudes require special provision for reliability.

As an example, the spacecraft *Cassini–Huygens*, launched in 1997, contained two identical flight recorders, each with 2.5 gigabits of memory in the form of arrays of commercial DRAM chips. Thanks to built-in EDAC functionality, spacecraft's engineering telemetry reported the number of (correctable) single-bit-per-word errors and (uncorrectable) double-bit-per-word errors. During the first 2.5 years of flight, the spacecraft reported a nearly constant single-bit error

rate of about 280 errors per day. However, on November 6, 1997, during the first month in space, the number of errors increased by more than a factor of four for that single day. This was attributed to a solar particle event that had been detected by the satellite GOES 9.[5]

There was some concern that as DRAM density increases further, and thus the components on chips get smaller, while at the same time operating voltages continue to fall, DRAM chips will be affected by such radiation more frequently—since lower-energy particles will be able to change a memory cell's state.[4] On the other hand, smaller cells make smaller targets, and moves to technologies such as SOI may make individual cells less susceptible and so counteract, or even reverse, this trend. Recent studies[6] show that single event upsets due to cosmic radiation have been dropping dramatically with process geometry and previous concerns over increasing bit cell error rates are unfounded.

Work published between 2007 and 2009 showed widely varying error rates with over 7 orders of magnitude difference, ranging from 10^{-10} error/bit·h (roughly one bit error per hour per gigabyte of memory) to 10^{-17} error/bit·h (roughly one bit error per millennium per gigabyte of memory).[6][7][8] A large-scale study based on Google's very large number of servers was presented at the SIGMETRICS/Performance'09 conference.[7] The actual error rate found was several orders of magnitude higher than the previous small-scale or laboratory studies, with between 25,000 (roughly 2.5×10^{-11} error/bit·h) and 70,000 (roughly 7×10^{-11} error/bit·h, or 5 bit errors per 8 gigabytes of RAM per hour) errors per billion device hours per megabit. More than 8% of DIMM memory modules were affected by errors per year.

The consequence of a memory error is system-dependent. In systems without ECC, an error can lead either to a crash or to corruption of data; in large-scale production sites, memory errors are one of the most common hardware causes of machine crashes.[7] Memory errors can cause security vulnerabilities.[7] A memory error can have no consequences if it changes a bit which neither causes observable malfunctioning nor affects data used in calculations or saved. A 2010 simulation study showed that, for a web browser, only a small fraction of memory errors caused data corruption, although, as many memory errors are intermittent and correlated, the effects of memory errors were greater than would be expected for independent soft errors.[9]

Some tests conclude that the isolation of DRAM memory cells can be circumvented by unintended side effects of specially crafted accesses to adjacent cells. Thus, accessing data stored in DRAM causes memory cell to leak their charges and interact electrically, as a result of high cell density in modern memory, altering the content of nearby memory rows that actually were not addressed in the original memory access. This effect is known as row hammer, and it has also been used in some privilege escalation computer security exploits.[10][11]

An example of a single-bit error that would be ignored by a system with no error-checking, would halt a machine with parity checking, or would be invisibly corrected by ECC: a single bit is stuck at 1 due to a faulty chip, or becomes changed to 1 due to background or cosmic radiation; a spreadsheet storing numbers in ASCII format is loaded, and the character "8" (decimal value 56 in the ASCII encoding) is stored in the byte that contains the stuck bit at its lowest bit position; then, a change is made to the spreadsheet and it is saved. As a result, the "8" (0011 100**0** binary) has silently become a "9" (0011 100**1**).

Solutions

Several approaches have been developed to deal with unwanted bit-flips, including immunity-aware programming, RAM parity memory, and ECC memory.

This problem can be mitigated by using DRAM modules that include extra memory bits and memory controllers that exploit these bits. These extra bits are used to record parity or to use an error-correcting code (ECC). Parity allows the detection of all single-bit errors (actually, any odd number of wrong bits). The most common error correcting code, a

single-error correction and double-error detection (SECDED) Hamming code, allows a single-bit error to be corrected and (in the usual configuration, with an extra parity bit) double-bit errors to be detected. Chipkill ECC is a more effective version that also corrects for multiple bit errors, including the loss of an entire memory chip.

Implementations

Seymour Cray famously said "parity is for farmers" when asked why he left this out of the CDC 6600.[12] Later, he included parity in the CDC 7600, which caused pundits to remark that "apparently a lot of farmers buy computers". The original IBM PC and all PCs until the early 1990s used parity checking.[13] Later ones mostly did not. Many current microprocessor memory controllers, including almost all AMD 64-bit offerings, support ECC, but many motherboards and in particular those using low-end chipsets do not.

An ECC-capable memory controller can detect and correct errors of a single bit per 64-bit "word" (the unit of bus transfer), and detect (but not correct) errors of two bits per 64-bit word. The BIOS in some computers, when matched with operating systems such as some versions of Linux, macOS, and Windows, allows counting of detected and corrected memory errors, in part to help identify failing memory modules before the problem becomes catastrophic.

Some DRAM chips include "internal" on-chip error correction circuits, which allow systems with non-ECC memory controllers to still gain most of the benefits of ECC memory.[14][15] In some systems, a similar effect may be achieved by using EOS memory modules.

Error detection and correction (EDAC) depends on an expectation of the kinds of errors that occur. Implicitly, it is assumed that the failure of each bit in a word of memory is independent, resulting in improbability of two simultaneous errors. This used to be the case when memory chips were one-bit wide, what was typical in the first half of the 1980s; later developments moved many bits into the same chip. This weakness is addressed by various technologies, including IBM's Chipkill, Sun Microsystems' Extended ECC, Hewlett Packard's Chipspare, and Intel's Single Device Data Correction (SDDC).

DRAM memory may provide increased protection against soft errors by relying on error correcting codes. Such error-correcting memory, known as *ECC* or *EDAC-protected* memory, is particularly desirable for high fault-tolerant applications, such as servers, as well as deep-space applications due to increased radiation. Some systems also "scrub" the memory, by periodically reading all addresses and writing back corrected versions if necessary to remove soft errors.

Interleaving allows for distribution of the effect of a single cosmic ray, potentially upsetting multiple physically neighboring bits across multiple words by associating neighboring bits to different words. As long as a single event upset (SEU) does not exceed the error threshold (e.g., a single error) in any particular word between accesses, it can be corrected (e.g., by a single-bit error correcting code), and an effectively error-free memory system may be maintained.[16]

Error-correcting memory controllers traditionally use Hamming codes, although some use triple modular redundancy (TMR). The latter is preferred because its hardware is faster than that of Hamming error correction scheme.[16] Space satellite systems often use TMR,[17][18][19] although satellite RAM usually uses Hamming error correction.[20]

Many early implementations of ECC memory mask correctable errors, acting "as if" the error never occurred, and only report uncorrectable errors. Modern implementations log both correctable errors (CE) and uncorrectable errors (UE). Some people proactively replace memory modules that exhibit high error rates, in order to reduce the likelihood of uncorrectable error events.[21]

Many ECC memory systems use an "external" EDAC circuit between the CPU and the memory. A few systems with ECC memory use both internal and external EDAC systems; the external EDAC system should be designed to correct certain errors that the internal EDAC system is unable to correct.[14] Modern desktop and server CPUs integrate the EDAC circuit into the CPU,[22] especially with the shift toward CPU-integrated memory controllers, which are related to the NUMA architecture.

As of 2009, the most common error-correction codes use Hamming or Hsiao codes that provide single-bit error correction and double-bit error detection (SEC-DED). Other error-correction codes have been proposed for protecting memory – double-bit error correcting and triple-bit error detecting (DEC-TED) codes, single-nibble error correcting and double-nibble error detecting (SNC-DND) codes, Reed–Solomon error correction codes, etc. However, in practice, multi-bit correction is usually implemented by interleaving multiple SEC-DED codes.[23][24]

Early research attempted to minimize the area and delay overheads of ECC circuits. Hamming first demonstrated that SEC-DED codes were possible with one particular check matrix. Hsiao showed that an alternative matrix with odd weight columns provides SEC-DED capability with less hardware area and shorter delay than traditional Hamming SEC-DED codes. More recent research also attempts to minimize power in addition to minimizing area and delay.[25][26][27]

Cache

Many processors use error-correction codes in the on-chip cache, including the Intel Itanium and Xeon[28] processors, the AMD Athlon, Opteron, all Zen-[29] and Zen+-based[30] processors (EPYC, EPYC Embedded, Ryzen and Ryzen Threadripper), and the DEC Alpha 21264.[23][31]

As of 2006, EDC/ECC and ECC/ECC are the two most common cache error-protection techniques used in commercial microprocessors. The EDC/ECC technique uses an error-detecting code (EDC) in the level 1 cache. If an error is detected, data is recovered from ECC-protected level 2 cache. The ECC/ECC technique uses an ECC-protected level 1 cache and an ECC-protected level 2 cache.[32] CPUs that use the EDC/ECC technique always write-through all STOREs to the level 2 cache, so that when an error is detected during a read from the level 1 data cache, a copy of that data can be recovered from the level 2 cache.

Registered memory

Registered, or buffered, memory is not the same as ECC; these strategies perform different functions. It is usual for memory used in servers to be both registered, to allow many memory modules to be used without electrical problems, and ECC, for data integrity. Memory used in desktop computers is neither, for economy. However, unbuffered (not-registered) ECC memory is available,[33] and some non-server motherboards support ECC functionality of such modules when used with a CPU that supports ECC.[34] Registered memory does not work reliably in motherboards without buffering circuitry, and vice versa.

Two 8 GB DDR4-2133 ECC 1.2 V RDIMMs

Advantages and disadvantages

Ultimately, there is a trade-off between protection against unusual loss of data, and a higher cost.

ECC protects against undetected memory data corruption, and is used in computers where such corruption is unacceptable, for example in some scientific and financial computing applications, or in file servers. ECC also reduces the number of crashes that are especially unacceptable in multi-user server applications and maximum-availability systems. Most motherboards and processors for less critical application are not designed to support ECC so their prices can be kept lower. Some ECC-enabled boards and processors are able to support unbuffered (unregistered) ECC, but will also work with non-ECC memory; system firmware enables ECC functionality if the ECC RAM is installed.

ECC memory usually involves a higher price when compared to non-ECC memory, due to additional hardware required for producing ECC memory modules, and due to lower production volumes of ECC memory and associated system hardware. Motherboards, chipsets and processors that support ECC may also be more expensive.

ECC may lower memory performance by around 2–3 percent on some systems, depending on the application and implementation, due to the additional time needed for ECC memory controllers to perform error checking.[35] However, modern systems integrate ECC testing into the CPU, generating no additional delay to memory accesses as long as no errors are detected.[22][36][37]

References

1. Werner Fischer. "RAM Revealed" (http://www.admin-magazine.com/Articles/RAM-Revealed). *admin-magazine.com*. Retrieved October 20, 2014.
2. "A survey of techniques for improving error-resilience of DRAM (https://dx.doi.org/10.1016/j.sysarc.2018.09.004)", JSA, 2018
3. Single Event Upset at Ground Level, Eugene Normand, Member, IEEE, Boeing Defense & Space Group, Seattle, WA 98124-2499 (https://web.archive.org/web/20131021190327/http://pdf.yuri.se/files/art/2.pdf)
4. "A Survey of Techniques for Modeling and Improving Reliability of Computing Systems (https://dx.doi.org/10.1109/TPDS.2015.2426179)", IEEE TPDS, 2015
5. Gary M. Swift and Steven M. Guertin. "In-Flight Observations of Multiple-Bit Upset in DRAMs". Jet Propulsion Laboratory (https://trs.jpl.nasa.gov/bitstream/handle/2014/15831/00-1594.pdf?sequence=1)
6. Borucki, "Comparison of Accelerated DRAM Soft Error Rates Measured at Component and System Level", 46th Annual International Reliability Physics Symposium, Phoenix, 2008, pp. 482–487
7. Schroeder, Bianca; Pinheiro, Eduardo; Weber, Wolf-Dietrich (2009). *DRAM Errors in the Wild: A Large-Scale Field Study* (http://www.cs.toronto.edu/~bianca/papers/sigmetrics09.pdf) (PDF). *SIGMETRICS/Performance*. ACM. ISBN 978-1-60558-511-6. Lay summary (http://www.zdnet.com/blog/storage/dram-error-rates-nightmare-on-dimm-street/638) – *ZDNet*.
8. "A Memory Soft Error Measurement on Production Systems" (http://www.ece.rochester.edu/~xinli/usenix07/).
9. Li, Huang; Shen, Chu (2010). ""A Realistic Evaluation of Memory Hardware Errors and Software System Susceptibility". Usenix Annual Tech Conference 2010" (http://www.cs.rochester.edu/~kshen/papers/usenix2010-li.pdf) (PDF).
10. Yoongu Kim; Ross Daly; Jeremie Kim; Chris Fallin; Ji Hye Lee; Donghyuk Lee; Chris Wilkerson; Konrad Lai; Onur Mutlu (2014-06-24). "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors" (http://users.ece.cmu.edu/~yoonguk/papers/kim-isca14.pdf) (PDF). *ece.cmu.edu*. IEEE. Retrieved 2015-03-10.
11. Dan Goodin (2015-03-10). "Cutting-edge hack gives super user status by exploiting DRAM weakness" (https://arstechnica.com/security/2015/03/cutting-edge-hack-gives-super-user-status-by-exploiting-dram-weakness/). Ars Technica. Retrieved 2015-03-10.
12. "CDC 6600" (http://research.microsoft.com/~GBell/craytalk/sld047.htm). Microsoft Research. Retrieved 2011-11-23.
13. "Parity Checking" (http://www.pcguide.com/ref/ram/errChecking-c.html). Pcguide.com. 2001-04-17. Retrieved 2011-11-23.

14. A. H. Johnston. "Space Radiation Effects in Advanced Flash Memories" (http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/13431/1/01-2369.pdf) Archived (https://web.archive.org/web/20160304220536/http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/13431/1/01-2369.pdf) 2016-03-04 at the Wayback Machine. NASA Electronic Parts and Packaging Program (NEPP). 2001.
15. "ECC DRAM – Intelligent Memory" (http://www.intelligentmemory.com/ECC-DRAM/). *intelligentmemory.com*. Retrieved 2014-12-23.
16. "Using StrongArm SA-1110 in the On-Board Computer of Nanosatellite" (https://web.archive.org/web/20111002152735/http://www.apmcsta.org/File/doc/Conferences/6th%20meeting/Chen%20Zhenyu.doc). Tsinghua Space Center, Tsinghua University, Beijing. Archived from the original (http://www.apmcsta.org/File/doc/Conferences/6th%20meeting/Chen%20Zhenyu.doc) on 2011-10-02. Retrieved 2009-02-16.
17. "Actel engineers use triple-module redundancy in new rad-hard FPGA" (http://mae.pennnet.com/Articles/Article_Display.cfm?ARTICLE_ID=111934). Military & Aerospace Electronics. Retrieved 2009-02-16.
18. "SEU Hardening of Field Programmable Gate Arrays (FPGAs) For Space Applications and Device Characterization" (https://web.archive.org/web/20111125185602/http://klabs.org/richcontent/Papers/Synopses/nsrec94.htm). Klabs.org. 2010-02-03. Archived from the original (http://klabs.org/richcontent/Papers/Synopses/nsrec94.htm) on 2011-11-25. Retrieved 2011-11-23.
19. "FPGAs in Space" (http://www.techfocusmedia.net/fpgajournal/feature_articles/20040803_space.htm). Techfocusmedia.net. Retrieved 2011-11-23.
20. "Commercial Microelectronics Technologies for Applications in the Satellite Radiation Environment" (http://radhome.gsfc.nasa.gov/radhome/papers/aspen.htm). Radhome.gsfc.nasa.gov. Retrieved 2011-11-23.
21. Doug Thompson, Mauro Carvalho Chehab. "EDAC - Error Detection And Correction" (https://www.kernel.org/doc/Documentation/edac.txt) Archived (https://web.archive.org/web/20090905174616/http://www.kernel.org/doc/Documentation/edac.txt) 2009-09-05 at the Wayback Machine. 2005 - 2009. "The 'edac' kernel module goal is to detect and report errors that occur within the computer system running under linux."
22. "AMD-762™ System Controller Software/BIOS Design Guide, p. 179" (http://support.amd.com/us/ChipsetMotherboard_TechDocs/24462.pdf) (PDF).
23. Doe Hyun Yoon; Mattan Erez. "Memory Mapped ECC: Low-Cost Error Protection for Last Level Caches" (http://users.ece.utexas.edu/~merez/mme_isca09.pdf). 2009. p. 3
24. Daniele Rossi; Nicola Timoncini; Michael Spica; Cecilia Metra. "Error Correcting Code Analysis for Cache Memory High Reliability and Performance" (http://www.date-conference.com/proceedings/PAPERS/2011/DATE11/PDFFILES/12.5_3.PDF) Archived (https://web.archive.org/web/20150203093300/http://www.date-conference.com/proceedings/PAPERS/2011/DATE11/PDFFILES/12.5_3.PDF) 2015-02-03 at the Wayback Machine.
25. Shalini Ghosh; Sugato Basu; and Nur A. Touba. "Selecting Error Correcting Codes to Minimize Power in Memory Checker Circuits" (https://hostdb.ece.utexas.edu/~touba/research/jlpe05.pdf) Archived (https://web.archive.org/web/20150203100831/https://hostdb.ece.utexas.edu/~touba/research/jlpe05.pdf) 2015-02-03 at the Wayback Machine. p. 2 and p. 4.
26. Chris Wilkerson; Alaa R. Alameldeen; Zeshan Chishti; Wei Wu; Dinesh Somasekhar; Shih-lien Lu. "Reducing cache power with low-cost, multi-bit error-correcting codes" (http://dl.acm.org/citation.cfm?id=1815973). doi: 10.1145/1816038.1815973 (https://doi.org/10.1145%2F1816038.1815973).
27. M. Y. Hsiao. "A Class of Optimal Minimum Odd-weight-column SEC-DED Codes" (http://www.cs.berkeley.edu/~culler/cs252-s02/papers/hsiao70.pdf). 1970.
28. Intel Corporation. "Intel Xeon Processor E7 Family: Reliability, Availability, and Serviceability" (http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/xeon-e7-family-ras-server-paper.pdf). 2011. p. 12.
29. "AMD Zen microarchitecture - Memory Hierarchy" (https://en.wikichip.org/wiki/amd/microarchitectures/zen#Memory_Hierarchy). *WikiChip*. Retrieved 15 October 2018.
30. "AMD Zen+ microarchitecture - Memory Hierarchy" (https://en.wikichip.org/wiki/amd/microarchitectures/zen%2B#Memory_Hierarchy). *WikiChip*. Retrieved 15 October 2018.
31. Jangwoo Kim; Nikos Hardavellas; Ken Mai; Babak Falsafi; James C. Hoe. "Multi-bit Error Tolerant Caches Using Two-Dimensional Error Coding" (http://www.ece.cmu.edu/~truss/papers/micro07-multibit.pdf). 2007. p. 2.

32. Nathan N. Sadler and Daniel J. Sorin. "Choosing an Error Protection Scheme for a Microprocessor's L1 Data Cache" (http://people.ee.duke.edu/~sorin/papers/iccd06_perc.pdf). 2006. p. 1.
33. "Typical unbuffered ECC RAM module: Crucial CT25672BA1067" (http://www.crucial.com/uk/store/partspecs.aspx?IMODULE=CT25672BA1067).
34. Specification of desktop motherboard that supports both ECC and non-ECC unbuffered RAM with compatible CPUs (http://www.asus.com/Motherboards/AMD_AM3Plus/M5A78LUSB3/#specifications)
35. "Discussion of ECC on pcguide" (http://www.pcguide.com/ref/ram/errECC-c.html). Pcguide.com. 2001-04-17. Retrieved 2011-11-23.
36. Benchmark of AMD-762/Athlon platform with and without ECC (http://forum.buildyourown.org.uk/topic.asp?ARCHIVE=true&TOPIC_ID=16274) Archived (https://web.archive.org/web/20130615195425/http://forum.buildyourown.org.uk/topic.asp?ARCHIVE=true&TOPIC_ID=16274) 2013-06-15 at the Wayback Machine
37. "ECCploit: ECC Memory Vulnerable to Rowhammer Attacks After All" (https://www.vusec.net/projects/eccploit/). Systems and Network Security Group at VU Amsterdam. Retrieved 2018-11-22.

External links

- SoftECC: A System for Software Memory Integrity Checking (http://pdos.csail.mit.edu/papers/softecc:ddopson-meng/softecc_ddopson-meng.pdf)
- A Tunable, Software-based DRAM Error Detection and Correction Library for HPC (http://www.fiala.me/pubs/papers/libsdc11.pdf)
- Detection and Correction of Silent Data Corruption for Large-Scale High-Performance Computing (http://www.fiala.me/pubs/papers/sc12-redmpi.pdf)
- Single-Bit Errors: A Memory Module Supplier's perspective on cause, impact and detection (http://www.smartm.com/files/salesLiterature/dram/smart_whitepaper_sbe.pdf)
- Intel Xeon Processor E3 - 1200 Product Family Memory Configuration Guide (https://web.archive.org/web/20131228061309/http://cache-www.intel.com/cd/00/00/46/78/467819_467819.pdf)