Solution for Excessive Memory Overheads : RAIDDR ECC code
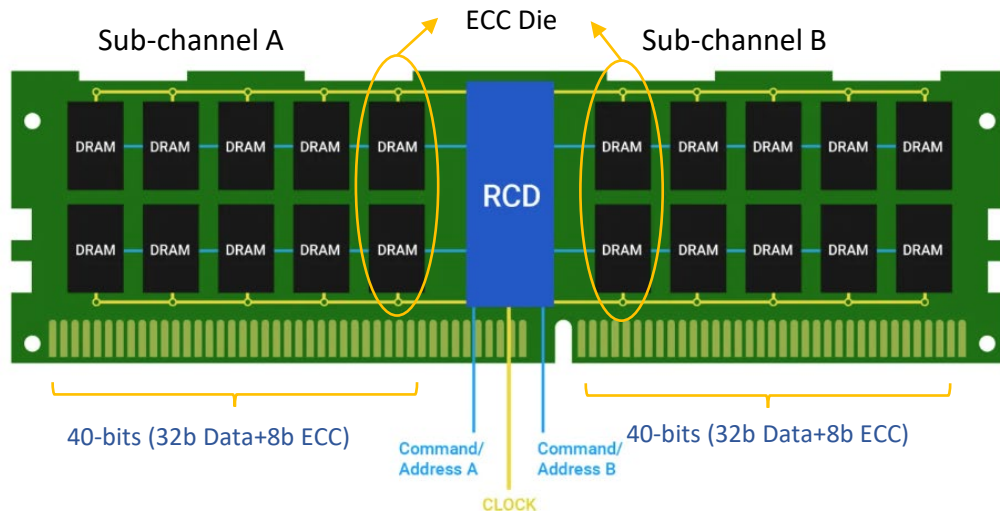
Terry Grunzke – Microsoft Azure

Agenda

❖ Overview of current Server Memory ECC and two significant issues

❖ Demonstration of how current memory ECC is disproportionately overprovisioned

❖ Explanation of how RAIDDR ECC code can address these issues and provide right-sized ECC provisioning with acceptable reliability

*SDDC - Single Device Data Correction (as known as Chipkill)

FMS
the Future of Memory and Storage

Terminologies: DDR5 DIMM Architecture

DDR5 10x4 DIMM

Sub-channel A   ECC Die   Sub-channel B

RCD

40-bits (32b Data+8b ECC)   40-bits (32b Data+8b ECC)

Command/Address A   Command/Address B

CLOCK
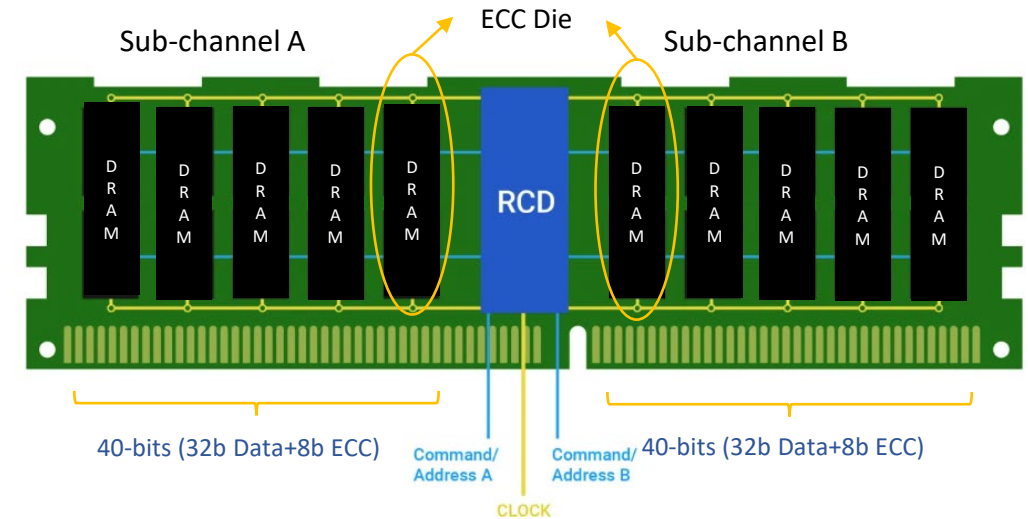
- ❖ x4 device has 4 DQs (Data I/Os)
- ❖ Each sub-channel consists of **8** DRAM placements for **data** and **2** for storing **ECC** bits
- ❖ Each sub-channel can transfer separate 512b of data and 128b of ECC with BL=16
- ❖ **Can support Chipkill*** and hence provides better reliability

DDR5 5x8 DIMM

Sub-channel A   ECC Die   Sub-channel B

RCD

40-bits (32b Data+8b ECC)   40-bits (32b Data+8b ECC)

Command/Address A   Command/Address B

CLOCK

- ❖ x8 device has 8 DQs (Data I/Os)
- ❖ Each sub-channel consists of **4** DRAM placements for **data** and **1** for storing **ECC** bits
- ❖ Each sub-channel can transfer separate 512b of data and 128b of ECC with BL=16
- ❖ **Can only support half Chipkill** due to availability of only one ECC die per sub-channel

*Chipkill protects memory even if one whole chip fails

FMS
the Future of Memory and Storage

Problem 1 : High ECC Overhead in DDR Memory

Costly Evolution of ECC in DDR

❖ DDR5 changed DIMM architecture (Enable deeper transaction)

    ❖ **DDR4** employs a structure of 2 ECC DRAM per 16 Data DRAM, resulting in an error correction **overhead** of **12.5%**.

    ❖ **DDR5** architecture requires 2 additional devices for ECC bits, leading to a total error correction **overhead** of **25%.**

❖ DDR5 DRAM added on-die SEC (Single Error Correct)

    ❖ Vendors concerned about single-bit errors as they continue to scale

    ❖ On-Die ECC **increased** DRAM die size by **~6%**

Total ECC Overhead and Correction Capability

❖ Total error correction overhead in DDR5 is **~31%**

    ❖ 2 ECC DRAM per 8 Data DRAM dies (**25%** overhead)

    ❖ Die size increase (**~6%** overhead for on-die ECC)

❖ Single-device correction capability between 100% and slightly less (depending on metadata usage))
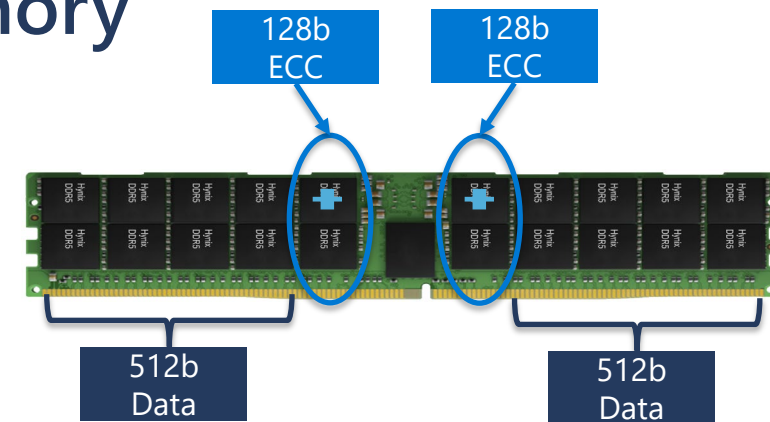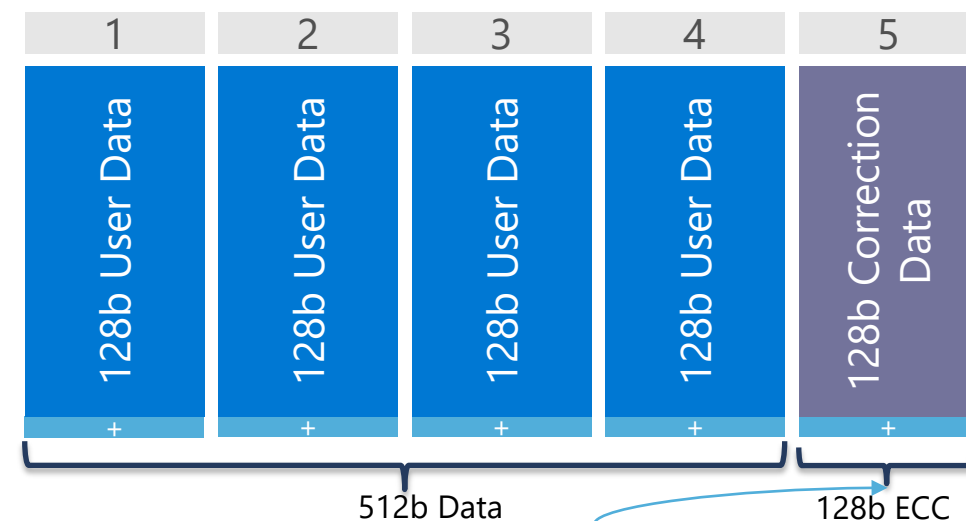

DDR5 10x4 Sub-channel Configuration
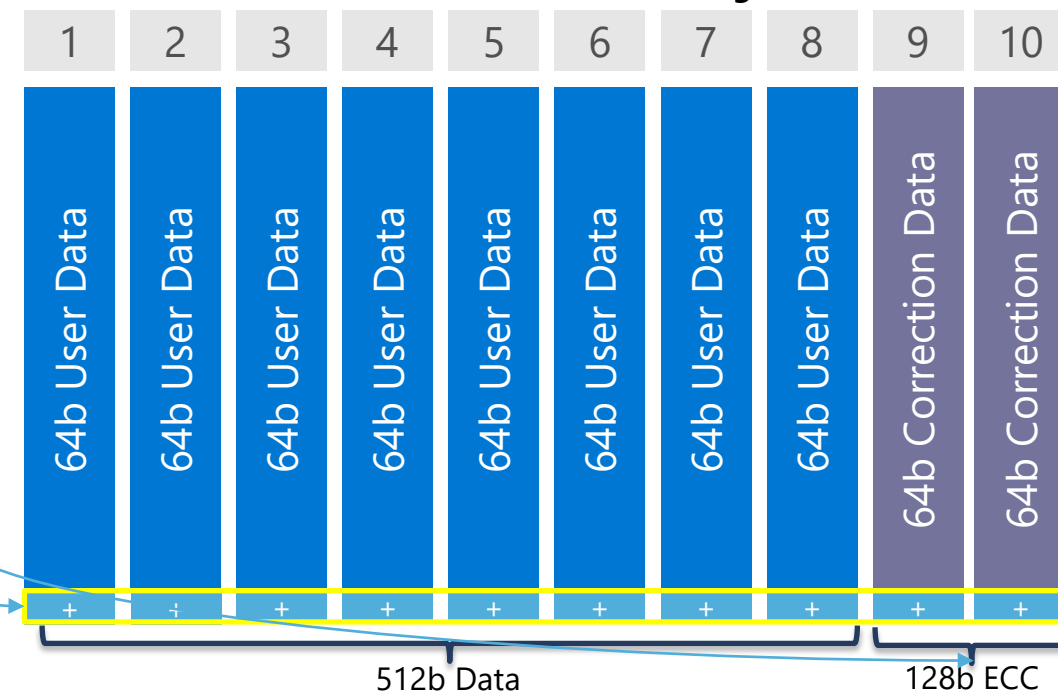
Key takeaway: DDR5 increased cost and power to improve bandwidth and maintain reliability

FMS
the **Future** of **Memory** and **Storage**
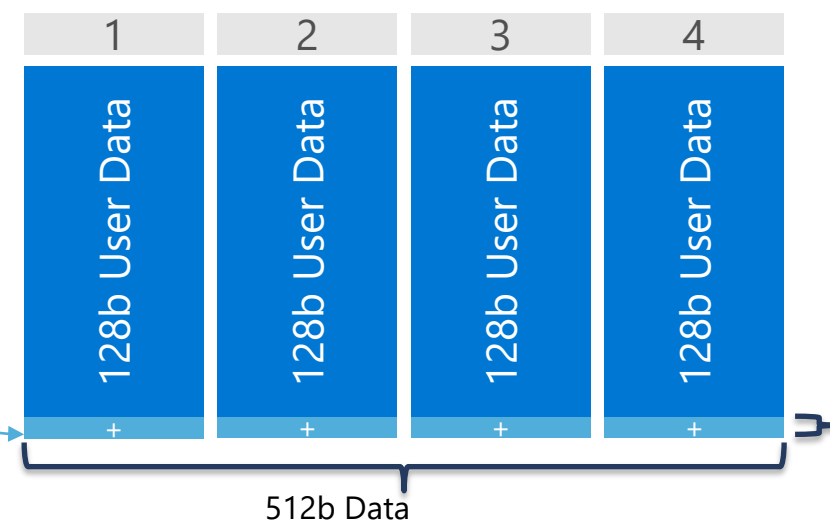
Problem 2 : High AFR/AIR of x8 devices

LPDDR5X/DDR5 x8

❖ DDR5/LPDDR5X **x8** would require **50% ECC overhead** to achieve chipkill. This is deemed an unacceptable cost adder.

  ❖ **DDR5** x8 architecture uses 1 additional device for ECC bits, leading to a total error correction **overhead of 25%, but with only ½ chipkill correction coverage.** This results in unacceptable AFR/AIR.

  ❖ **LPDDR5X x8 in servers** today does not provide any additional devices for ECC bits. This results in unacceptable AFR/AIR even with host access to on-die ECC bits .

  ❖ An additional **~6%** overhead due to die size increase does exist in both cases.

DDR5 5x8 Sub-channel Configuration

1	2	3	4	5
128b User Data	128b User Data	128b User Data	128b User Data	128b Correction Data

512b Data   128b ECC

LPDDR5X 4x8 Sub-channel Configuration (Grace)

1	2	3	4
128b User Data	128b User Data	128b User Data	128b User Data

512b Data

Key takeaway: LPDDR5X and DDR5 x8 architectures result in too high ECC overhead or unacceptable reliability

FMS
the Future of Memory and Storage

Properly provisioning ECC

How many extra bits is enough?

Two major memory fault classifications that both drive Memory AFR

1. **DRAM faults**, attributable to a malfunctioning DRAM → *Generally ECC correctable*

2. **Non-DRAM faults**, attributable to memory/ECC subsystem malfunction → *Generally not ECC correctable*

Focusing on improved ECC coverage for DRAM faults yields diminishing returns when DRAM-related AFR is below non-DRAM-related AFR

Current Enhanced RAIDDR proposals target ECC overhead of 1 DRAM + 32 additional ECC bits.

Memory AFR as a function of DRAM AFR and observed non-DRAM AFR

the Future of Memory and Storage
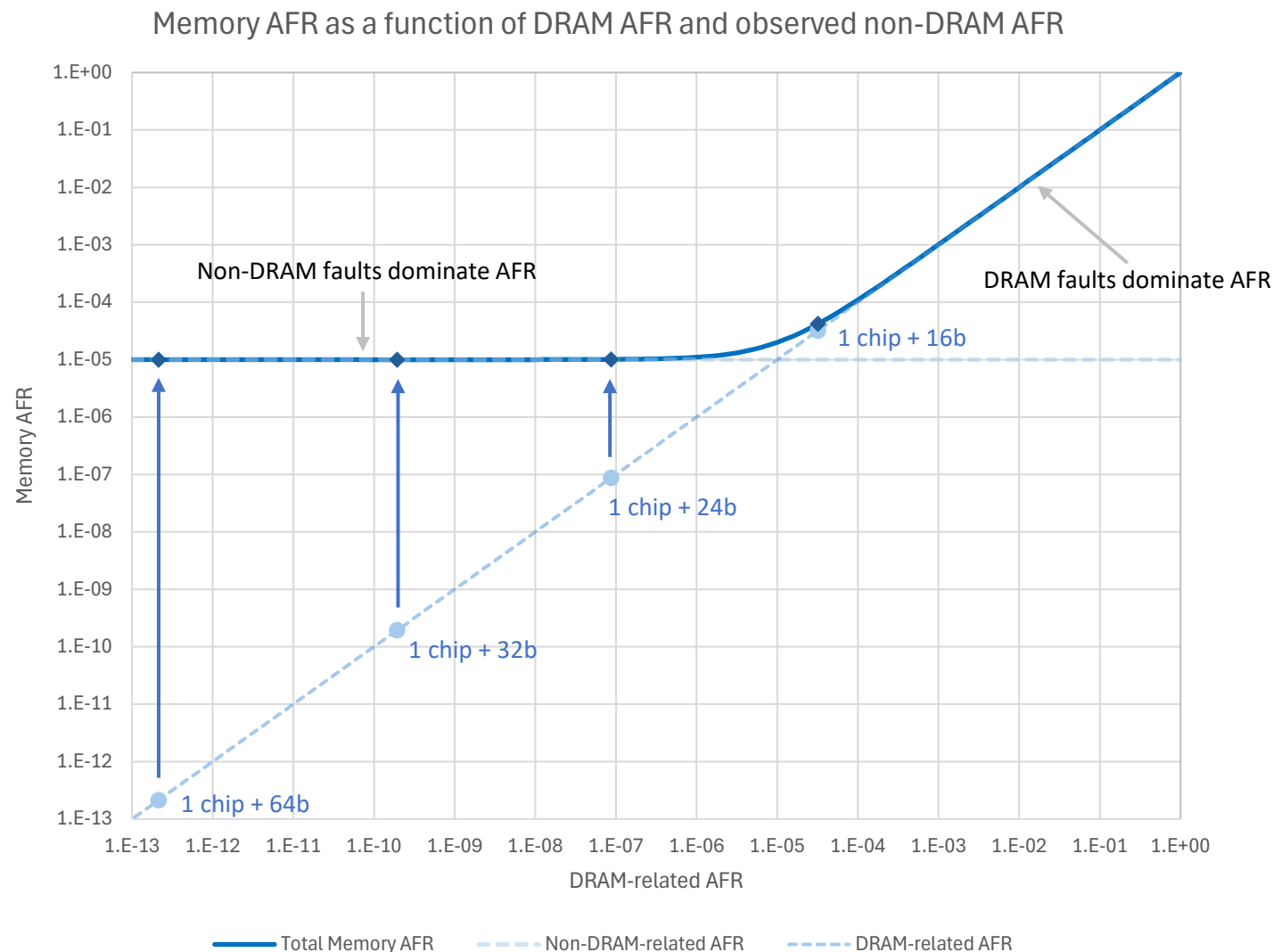
RAIDDR Objectives and Methods

Main Objectives

- ❖ Minimize ECC overhead to reduce DRAM placements
- ❖ Enable usage of x8 memory (DDR and LPDDR) with acceptable reliability (near Chipkill)
- ❖ Improve correctability with metadata usage

RAIDDR Scheme

- ❖ Provide additional bits to the host ECC.
- ❖ Remove 1 die placement per sub-channel to reduce per-DIMM cost and power. Similar to DDR5 10x4 to 9x4
- ❖ Enhance ECC capability to handle single-device failure plus one additional bit (SDDC*+1)

*SDDC - Single Device Data Correction (as known as Chipkill)

DRAM output extended to include additional bits

68b ECC

68b ECC

512b+32b Data

512b+32b Data

1	2	3	4	5	6	7	8	9	10
64b User Data	64b User Data	64b User Data	64b User Data	64b User Data	64b User Data	64b User Data	64b User Data	64b Correction Data	64b Correction Data

FMS
the Future of Memory and Storage

RAIDDR (RAID for DDR) - Introduction

- ❖ RAIDDR is a **symbol-based** error correction code developed by **Microsoft** and implemented in the memory controller.
  - ❖ A collection of bits including ECC bits (codeword) is divided into sections (**symbols**)
  - ❖ RAIDDR is just the code, the controller's use/implementation defines the requirements. Has two variants – Basic and Enhanced
- ❖ Basic RAIDDR
  - ❖ Similar to Reed-Solomon (RS), **this can correct errors in a single symbol** (SDDC*). Just the red bits in the diagram.
  - ❖ Very simple to implement, **better error detection than RS**
- ❖ Enhanced RAIDDR
  - ❖ More complicated to implement than Basic RAIDDR.
  - ❖ Provides correction of an **additional bit anywhere in the code** (SDDC+1b). The red and **orange** bits in the diagram.

DDR5 9x4 Sub-channel Configuration

| Die 1 | Die 2 | Die 3 | Die 4 | Die 5 | Die 6 | Die 7 | Die 8 | Parity |

- Basic RAIDDR is open license and available on GitHub: https://github.com/microsoft/BasicRAIDDR
- IP Vendors have implemented RAIDDR and shown that it delivers comparable or superior latency at a fraction the logic required for traditional schemes

*SDDC - Single Device Data Correction (as known as Chipkill)

the Future of Memory and Storage

Summary

- ❖ Today's servers either overprovision Memory ECC (e.g. DDR5 10x4) or do not have acceptable correction capabilities (e.g. LPDDR5/DDR5 x8)

- ❖ RAIDDR is a solution that can provide acceptable reliability for both problems

- ❖ Microsoft has provided RAIDDR with royalty free open license

*SDDC - Single Device Data Correction (as known as Chipkill)

the Future of Memory and Storage