Life Beyond Flash
New Non-Volatile Memory Technologies

Bill Gervasi

Principal Systems Architect

bilge@Nantero.com

Agenda

- Checkpointing
- Problem Statement
- Tiers of memory & storage
- A New Memory Specification
- Data Persistence

Data processing is great


Data processing is great
Until something goes wrong

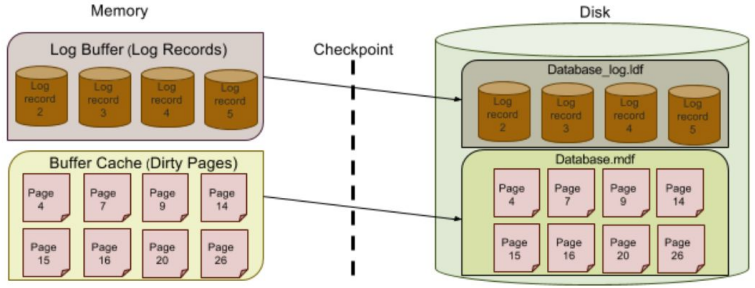
Checkpoint

November 12, 2015 Alexandr Omelchenko Glossary

[Total: 21 Average: 4.2/5]

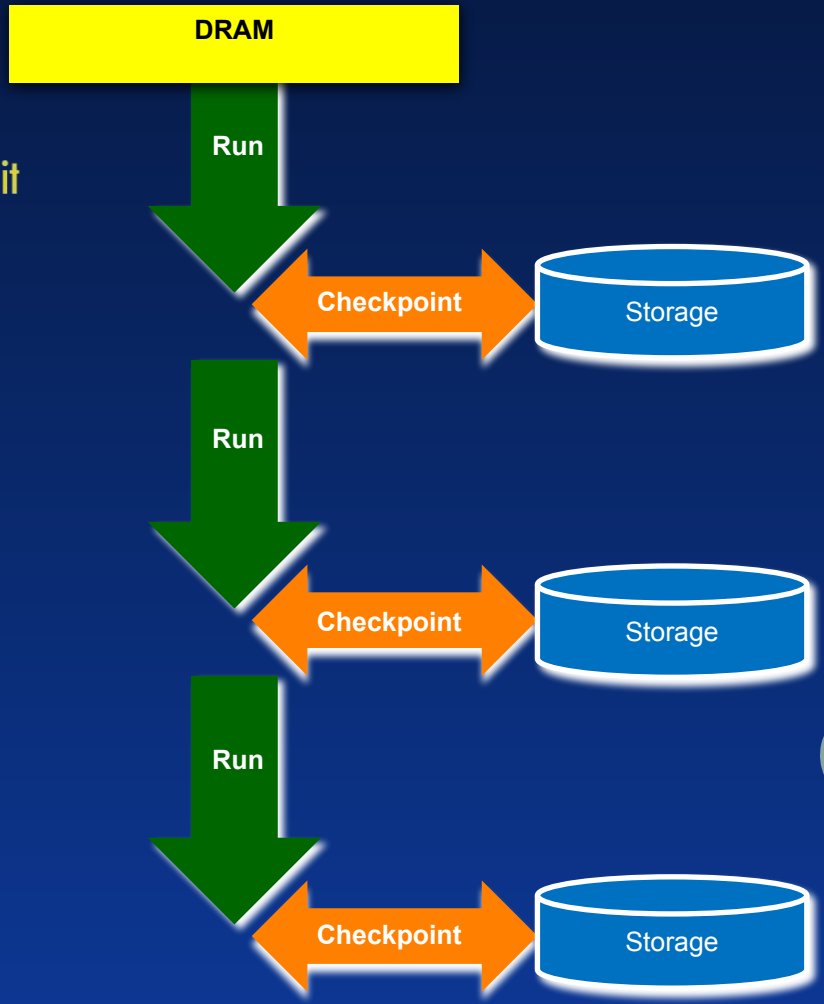
Checkpoint is a process that writes current in-memory dirty pages (modified pages) and transaction log records to physical disk. In SQL Server checkpoints are used to reduce the time required for recovery in the event of system failure. Checkpoint is regularly issued for each database. The following set of operations starts when checkpoint occurs:

1. Log records from log buffer (including the last log record) are written to the disk.
2. All dirty data file pages (pages that have been modified since the last checkpoint or since they were read from disk) are written into the data file from the buffer cache.
3. Checkpoint LSN is recorded in the database boot page.



Checkpointing invented to compensate for volatile nature of main memory

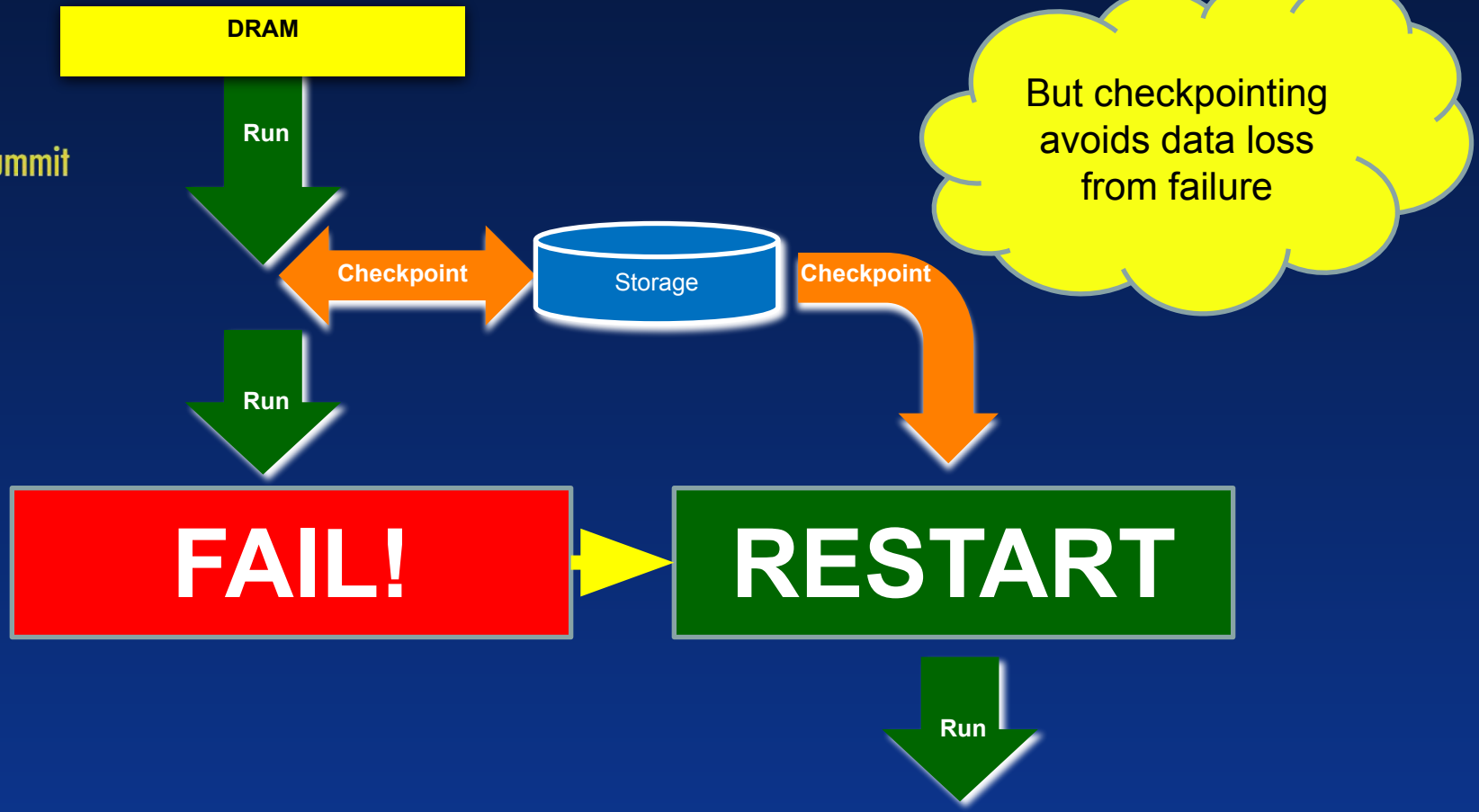
Save work in progress in case things crash



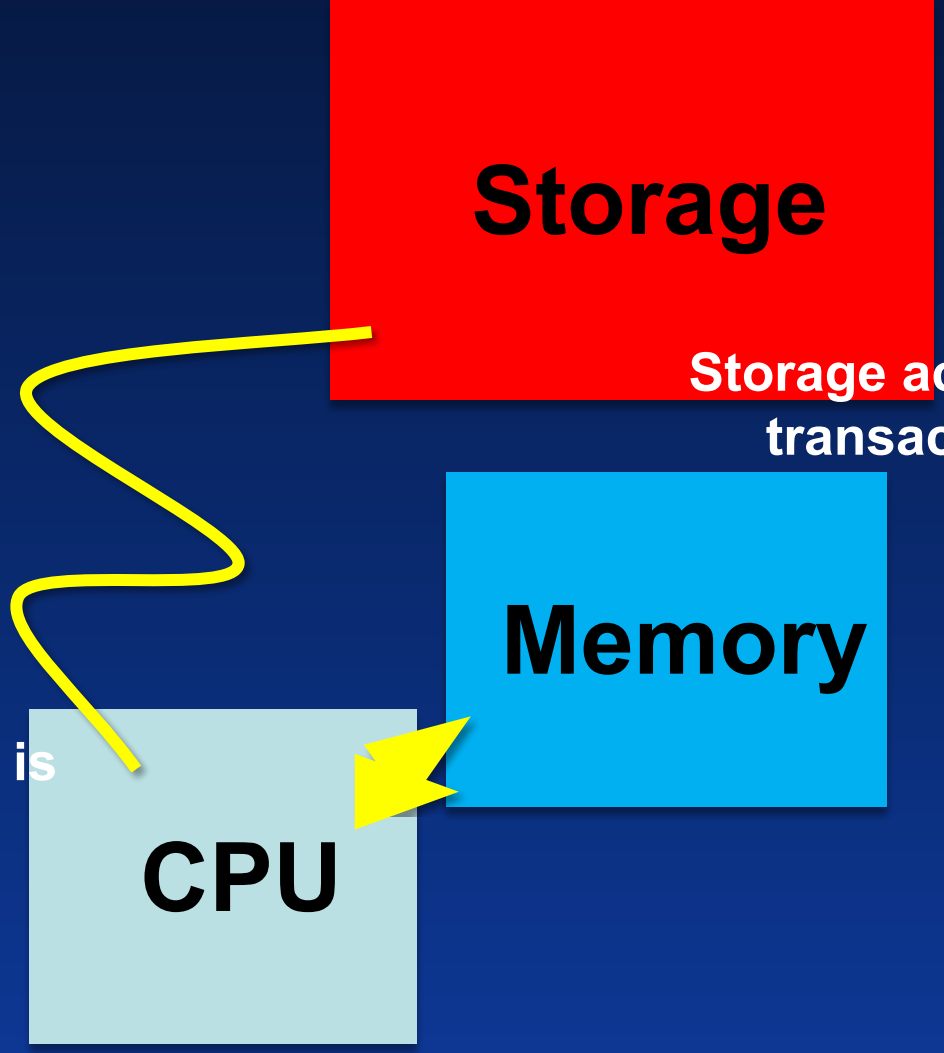
Checkpointing degrades performance

Checkpointing burns power

Checkpointing sucks



But checkpointing avoids data loss from failure

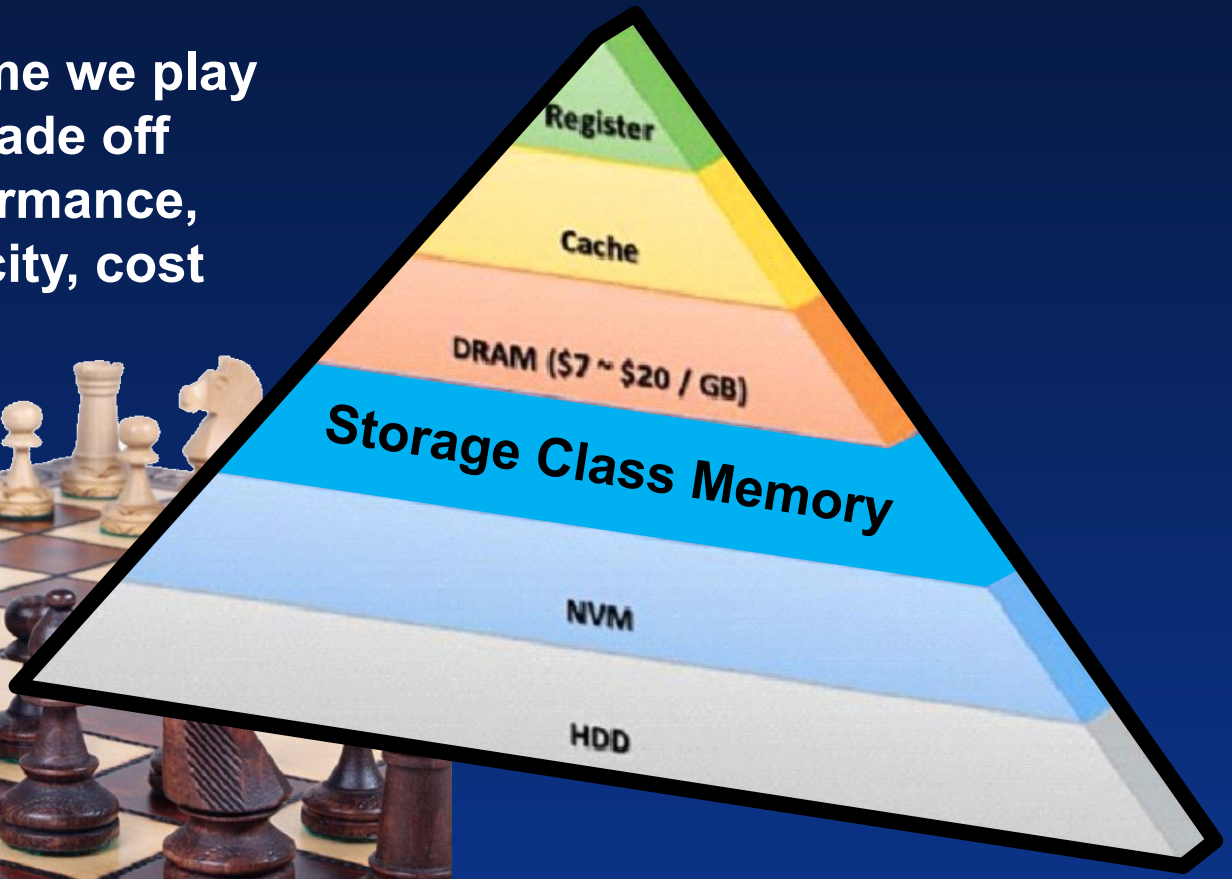


Storage access time impacts transaction granularity

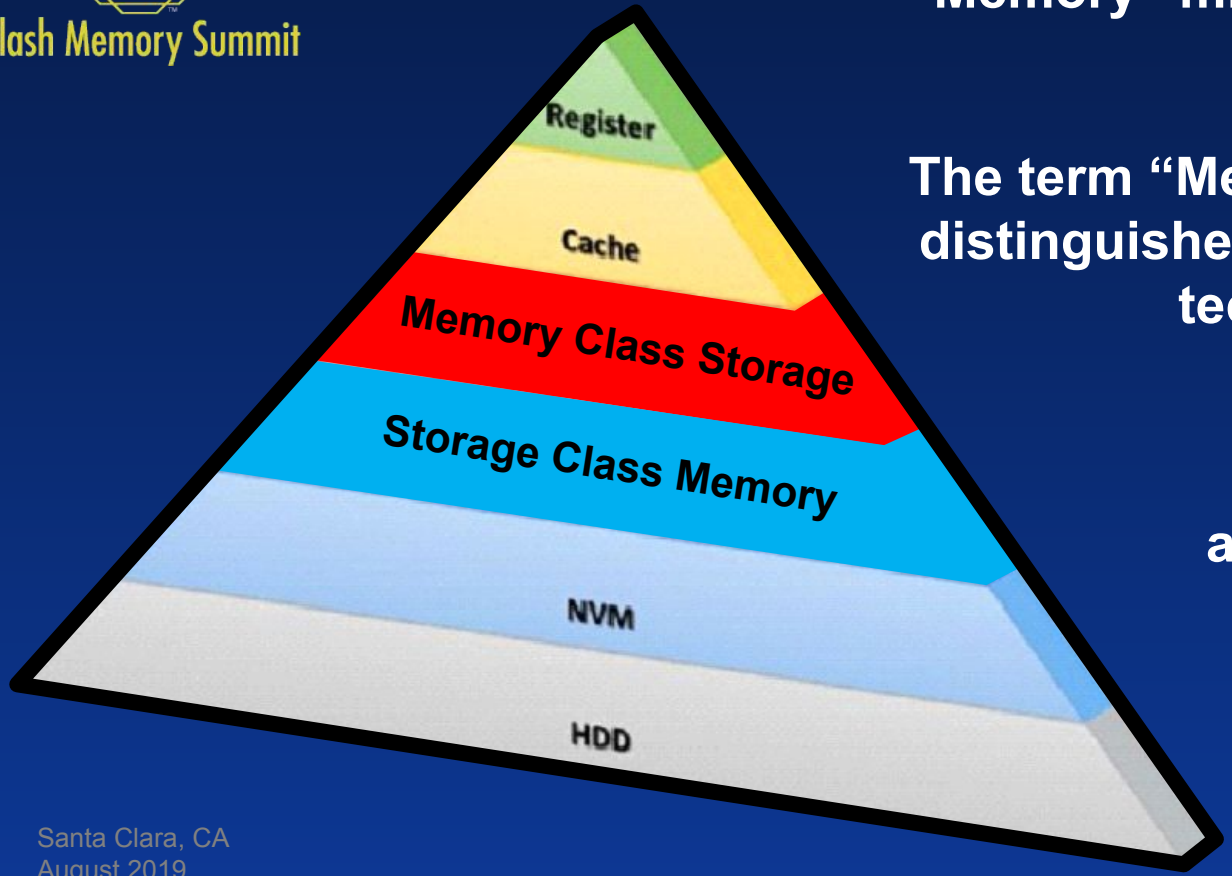
Data persistence is essential

Flash Memory Summit

The game we play to trade off performance, capacity, cost



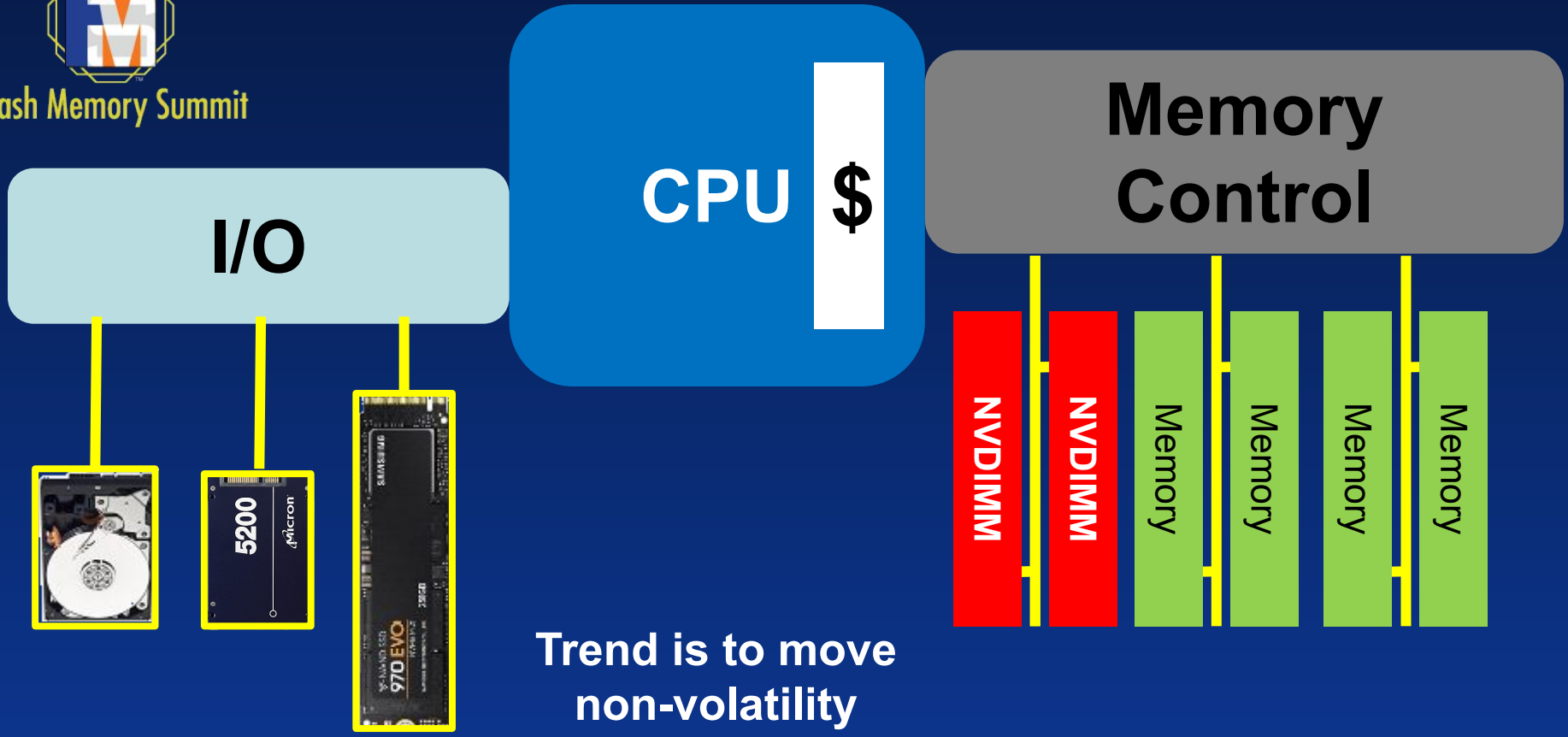
Flash Memory Summit



Renaming SCM as "Persistent Memory" missed the mark

The term "Memory Class Storage" distinguishes DRAM replacement technologies

MCS & SCM are both persistent, just have very different performances



Trend is to move non-volatility closer to the CPU

Flash Memory

The Search for

THE HOLY GRAIL

End to end…

Data Persistence

…but without giving up
PERFORMANCE

Memory Class Storage Checklist

Persistent

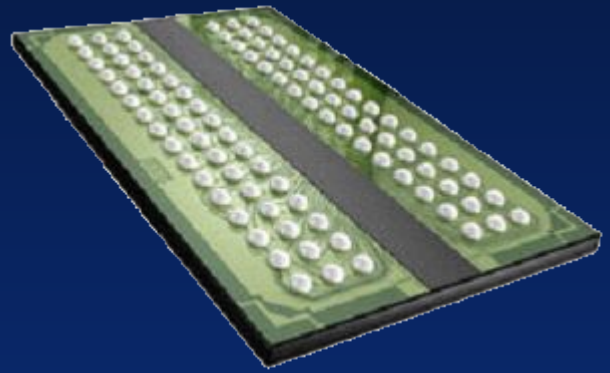
No compromise DRAM speed

Fully deterministic

No device wear out visible to the controller

Interface protocol compliant, e.g., DDR5

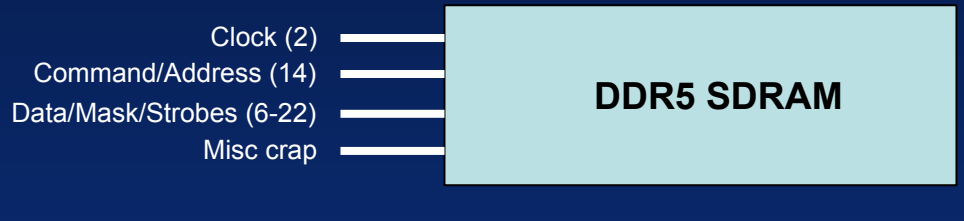
What exactly is DDR5, anyway?

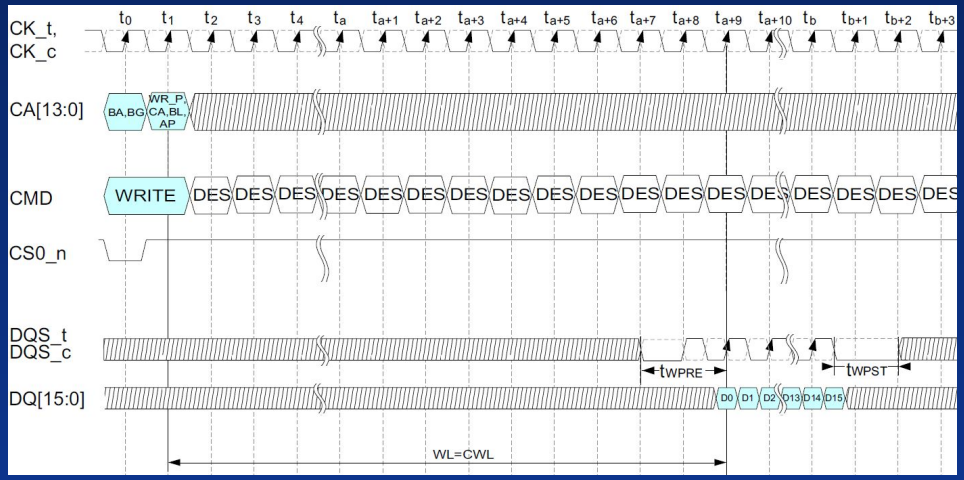


Signaling protocol for CPU to DRAM

Tradeoff of performance versus pin count

Designed to “just meet” target speeds





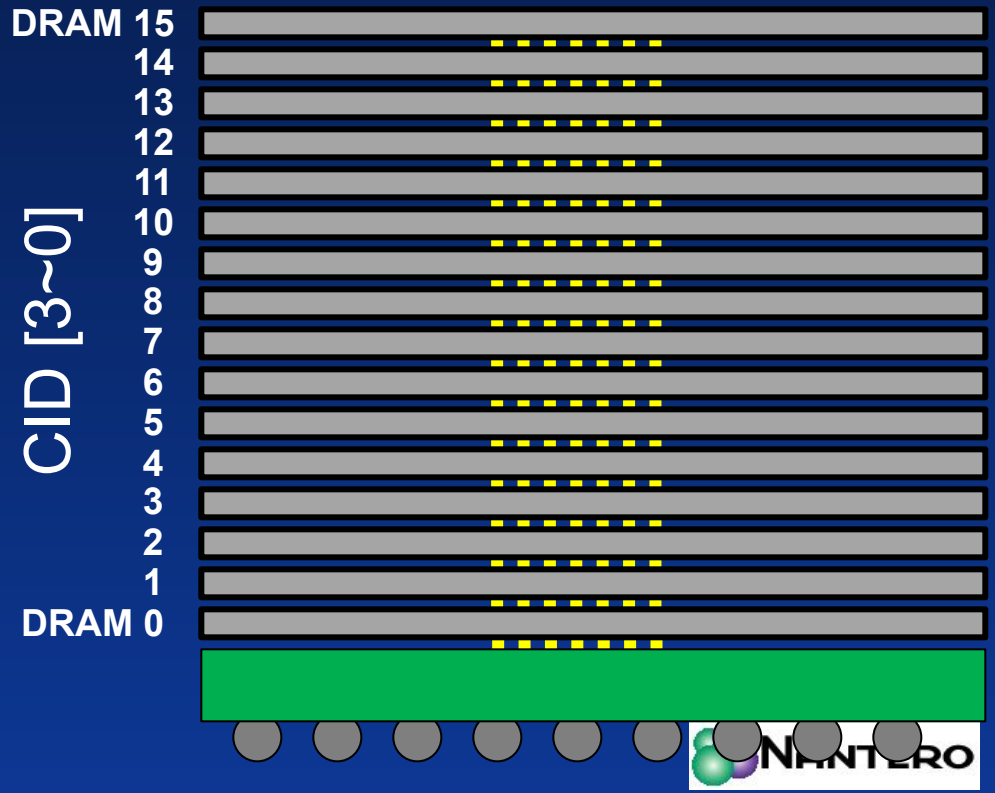
- Single-ended signals, differential clocks/strobes
- Bi-directional data bus
- Byte level data masking
- Deterministic timing from command to data
- ECC required on data
- Multiplexed address/command signals
- Some flexibility on data latencies

DDR5 SDRAM Core Functions

Activate	Fetches data from the array to sense amps
Read	Reads data from sense amps
Write	Writes data to sense amps
Precharge	Restores data from sense amps to the array
Refresh	Restores charge levels in the array
Self Refresh	Keeps data valid while device is idle

Chip ID

Selects one DRAM die within a stack of die in a package



8 BANK GROUPS WITH 4 BANKS PER GROUP = 32 BANKS PER DRAM DIE

Bank Group 7	6	5	4	3	2	1	Bank Group 0
BANK 3	3	3	3	3	3	3	BANK 3
BANK 2	2	2	2	2	2	2	BANK 2
BANK 1	1	1	1	1	1	1	BANK 1
BANK 0	0	0	0	0	0	0	BANK 0

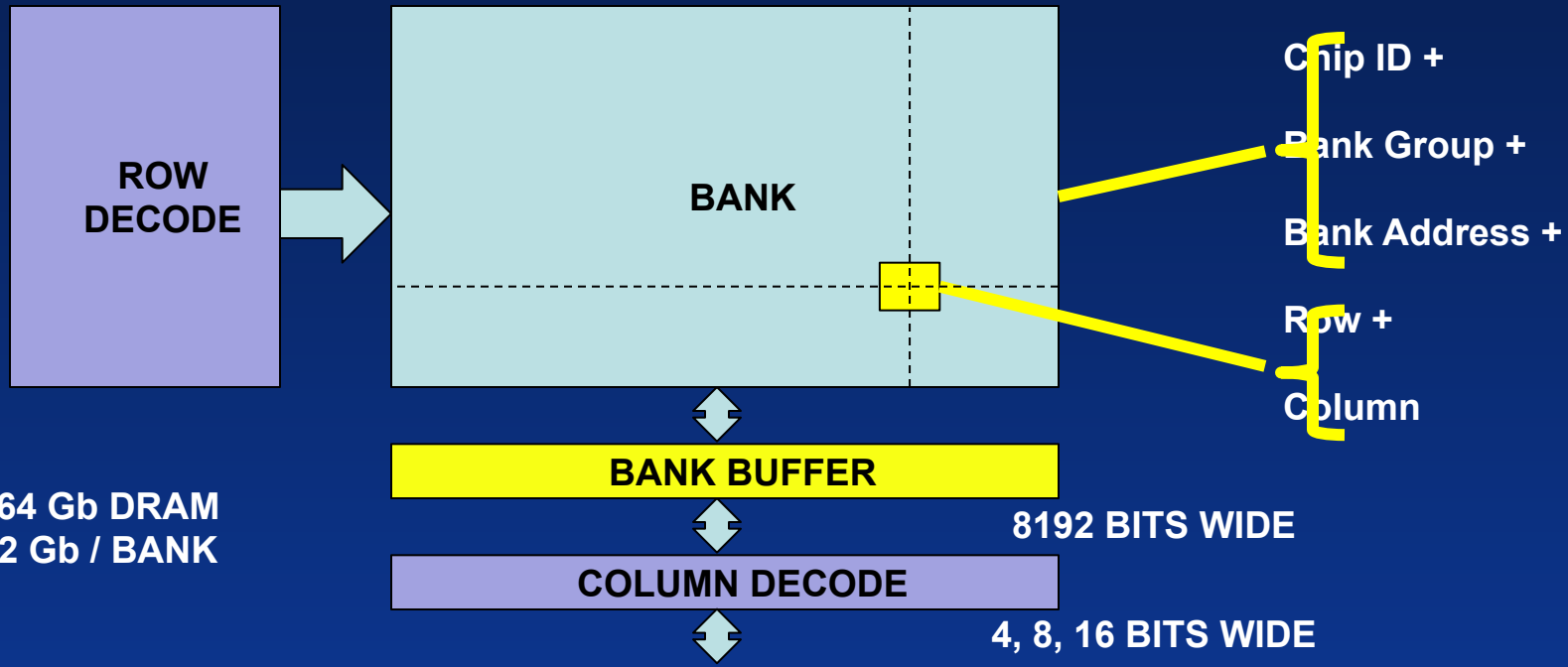

A specific bit is addressed by
ROW DECODE
BANK
Chip ID +
Bank Group +
Bank Address +
Row +
Column
BANK BUFFER
8192 BITS WIDE
COLUMN DECODE
4, 8, 16 BITS WIDE
64 Gb DRAM
2 Gb / BANK

Table 9 — 32 Gb Addressing Table

Configuration		8 Gb x4	4 Gb x8	2 Gb x16
Bank Address	BG Address	BG0~BG2	BG0~BG2	BG0~BG1
	Bank Address in a BG	BA0~BA1	BA0~BA1	BA0~BA1
	# BG / # Banks per BG / # Banks	8 / 4 / 32	8 / 4 / 32	4 / 4 / 16
Row Address		R0~R16	R0~R16	R0~R16
Column Address		C0~C10	C0~C9	C0~C9
Page size		1KB	1KB	2KB
Chip IDs / Maximum Stack Height		CID0~3 / 16H	CID0~3 / 16H	CID0~3 / 16H

Table 10 — 64 Gb Addressing Table

Configuration		16 Gb x4	8 Gb x8	4 Gb x16
Bank Address	BG Address	BG0~BG2	BG0~BG2	BG0~BG1
	Bank Address in a BG	BA0~BA1	BA0~BA1	BA0~BA1
	# BG / # Banks per BG / # Banks	8 / 4 / 32	8 / 4 / 32	4 / 4 / 16
Row Address		R0~R17	R0~R17	R0~R17
Column Address		C0~C10	C0~C9	C0~C9
Page size		1KB	1KB	2KB
Chip IDs / Maximum Stack Height		CID0~2 / 8H	CID0~2 / 8H	CID0~2 / 8H

CHIP ID 3 “STOLEN” TO MAKE R17

This specific excerpt highlights the upper limit of the DDR5 protocol

16H 3DS stack of 32 Gb die = 512 Gb per pkg

8H 3DS stack of 64 Gb die = 512 Gb per pkg

Function	Abbreviation	CS_n	CA Pins												
			CA0	CA1	CA2	CA3	CA4	CA5	CA6	CA7	CA8	CA9	CA10	CA11	CA12	CA13
Activate	ACT	L	L	L	R0	R1	R2	R3	BA0	BA1	BG0	BG1	BG2	CID0	CID1	CID2/ DDPID
		H	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	CID3/ R17
Write	WR	L	H	L	H	H	L	BL*=L	BA0	BA1	BG0	BG1	BG2	CID0	CID1	CID2/ DDPID
		H	V	C3	C4	C5	C6	C7	C8	C9	C10	V	AP=L	WR Partial=L	V	CID3
Read	RD	L	H	L	H	H	H	BL*=L	BA0	BA1	BG0	BG1	BG2	CID0	CID1	CID2/ DDPID
		H	C2	C3	C4	C5	C6	C7	C8	C9	C10	V	AP=L	V	V	CID3

All bit positions are consumed
CID3 must be used as R17 for 64 Gb die

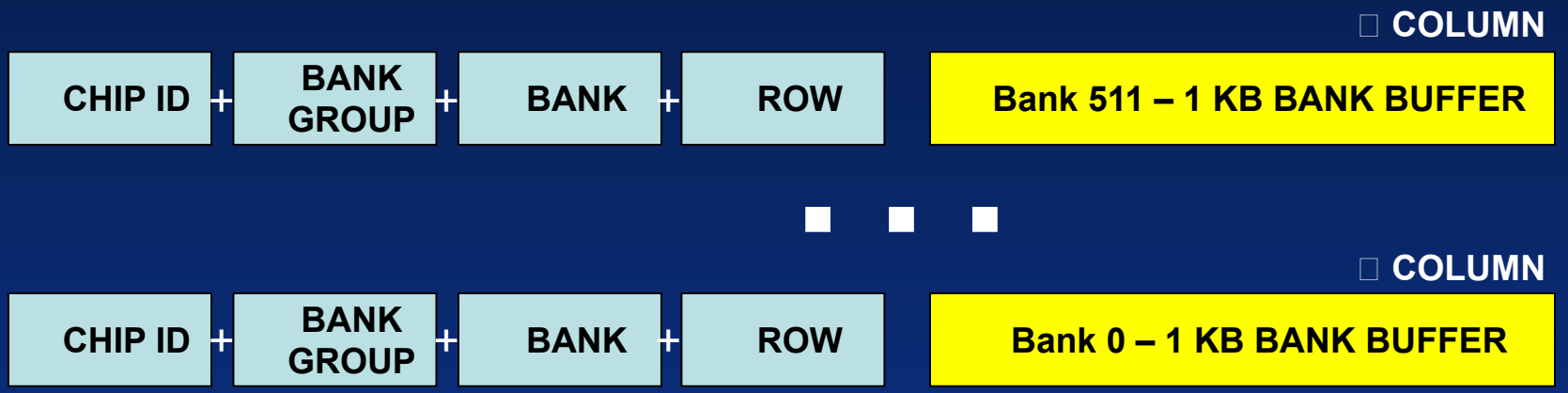
Per DRAM package…

Up to 16 DRAM die…

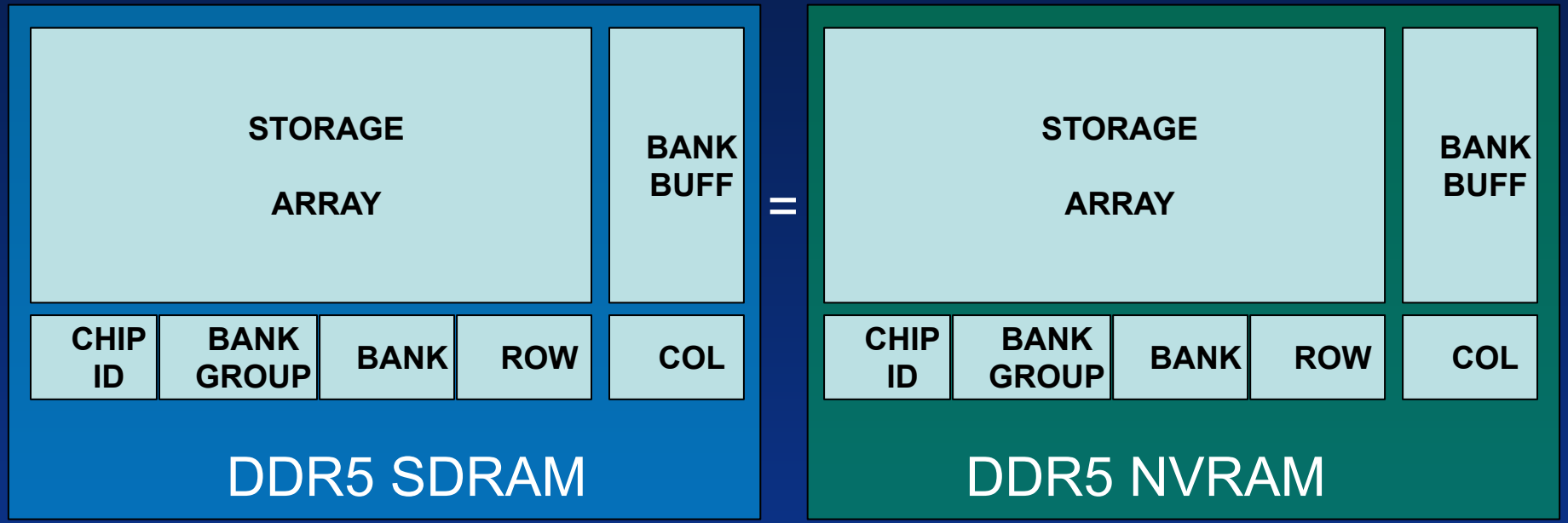
32 bank buffers…

1 KB per bank buffer…

512 KB max addressable at a time

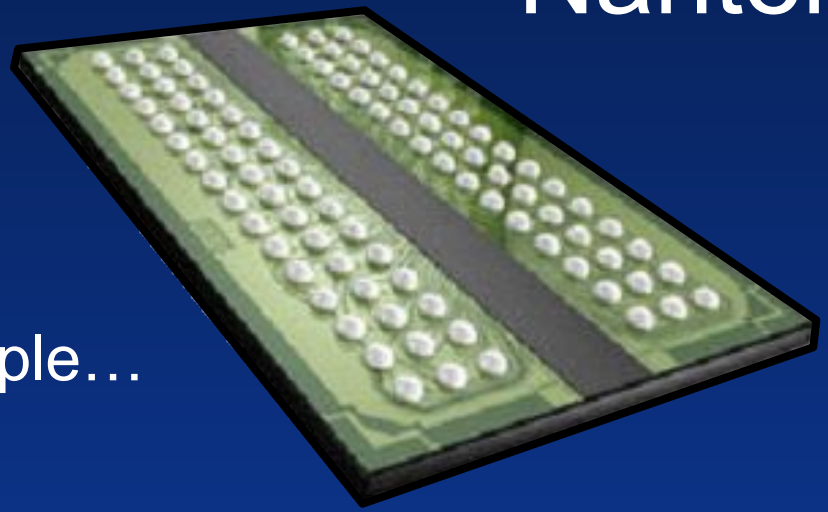


From controller perspective, each 1 KB block is associated with a specific combination of CID+BG+BA (9 bits) + ROW

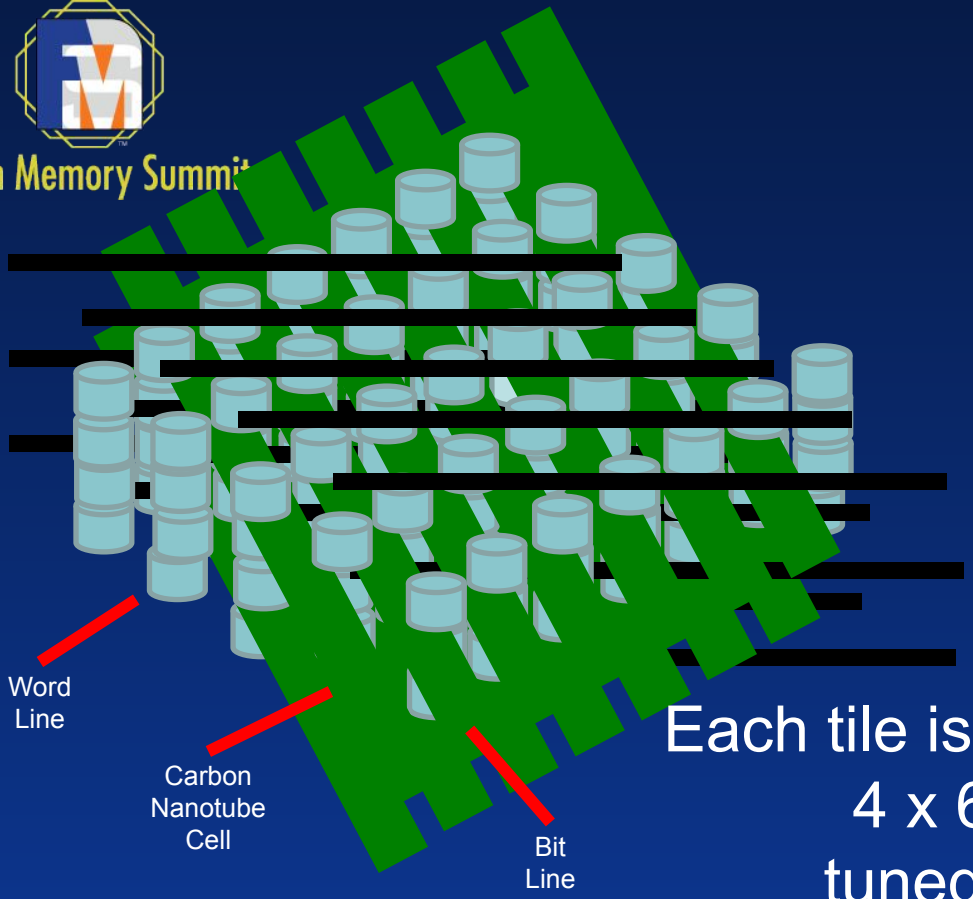


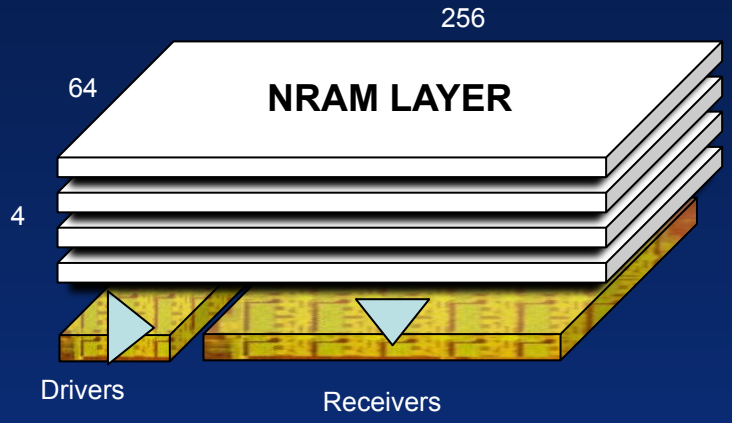
Any storage array may be abstracted

Nantero NRAM™



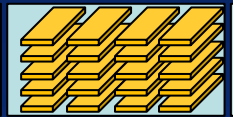
Oh, for example…



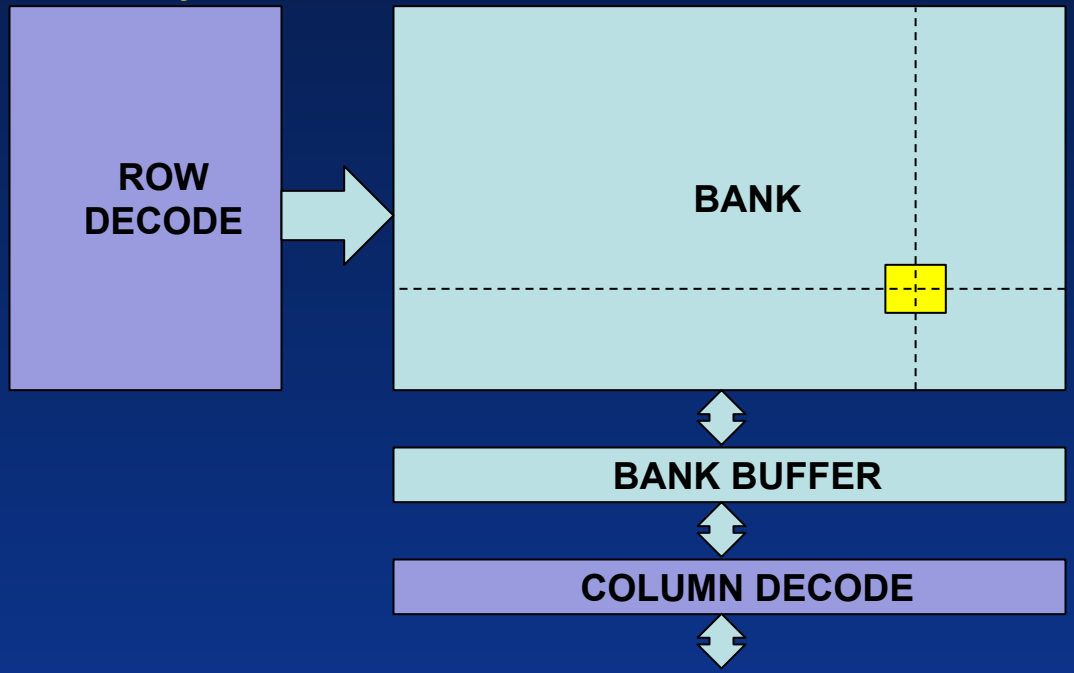


Each tile is a self-contained crosspoint
4 x 64 x 256 = 64 Kb unit
tuned to match DDR timing

64 Gb = 1024K Tiles = 32K Tiles per Bank

Bank Group 7	6	5	4	3	2	1	Bank Group 0
BANK 3		3	3	3	3	3	BANK 3
BANK 2	2	2	2	2	2	2	BANK 2
BANK 1	1	1	1	1	1	1	BANK 1
BANK 0	0	0	0	0	0	0	BANK 0

With an NVM core…



…Activates do not destroy contents

PRECHARGE operation is not needed

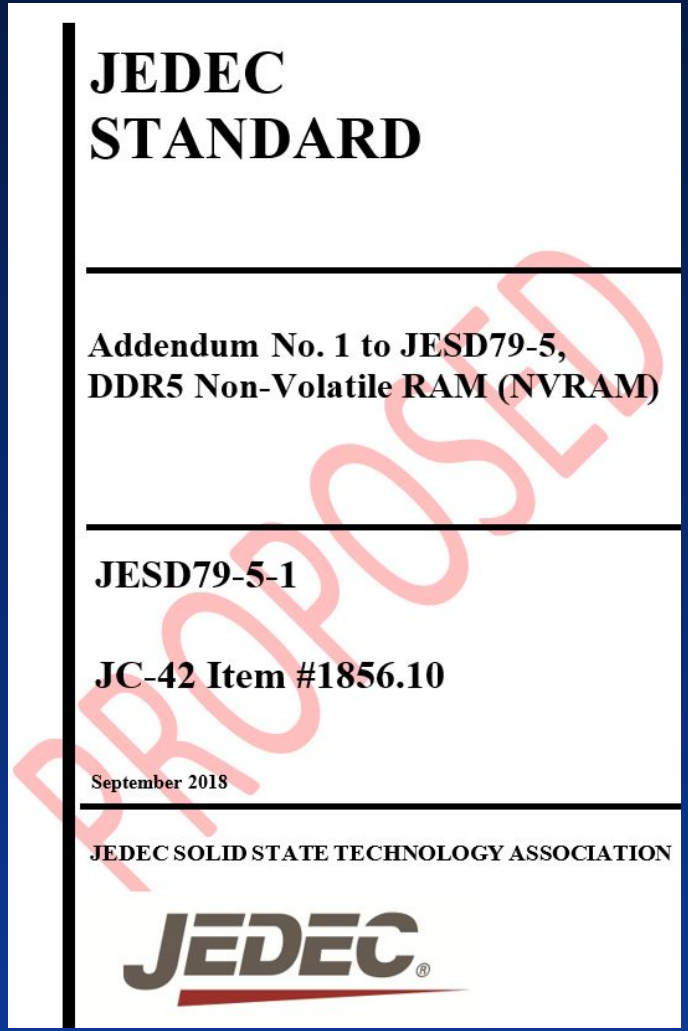
Banks never need to close

DDR5 SDRAM

SDRAM must close banks before reactivating
SDRAM must close banks prior to refreshing
Memory controller must buffer I/O data during refresh cycles

DDR5 NVRAM

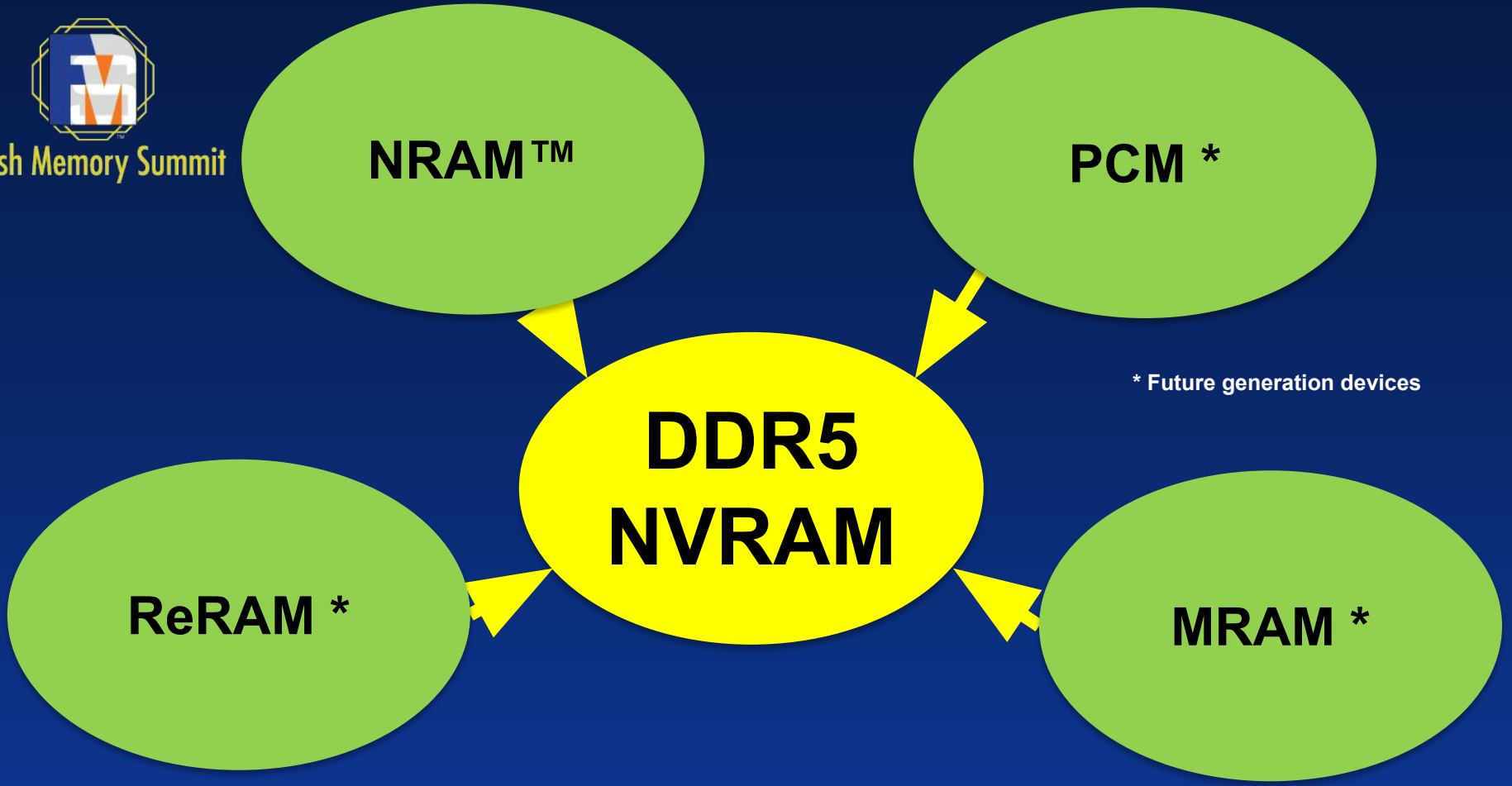
NVRAM is always available for data transfer
15% or more throughput at the same clock frequency



New specification in process

Addendum to DDR5 SDRAM

Describes differences


NRAM™
PCM *
DDR5 NVRAM
ReRAM *
MRAM *
* Future generation devices

Other persistent memory device types are emerging

They are also covered by the DDR5 NVRAM specification

Some possible differences from NRAM:

- Timing variations (activation, write recovery, etc.)
- Different page size per bank
- Destructive ACTIVATE (PRECHARGE needed)
- Persistence defined from FLUSH command

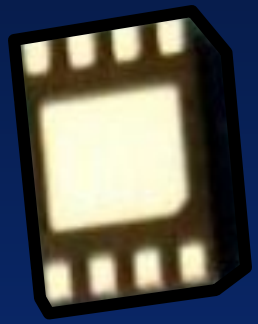
Common DDR5 NVRAM Feature Set

Profile 1 Features

Profile 2 Features

Profile 3 Features

Profile 4 Features

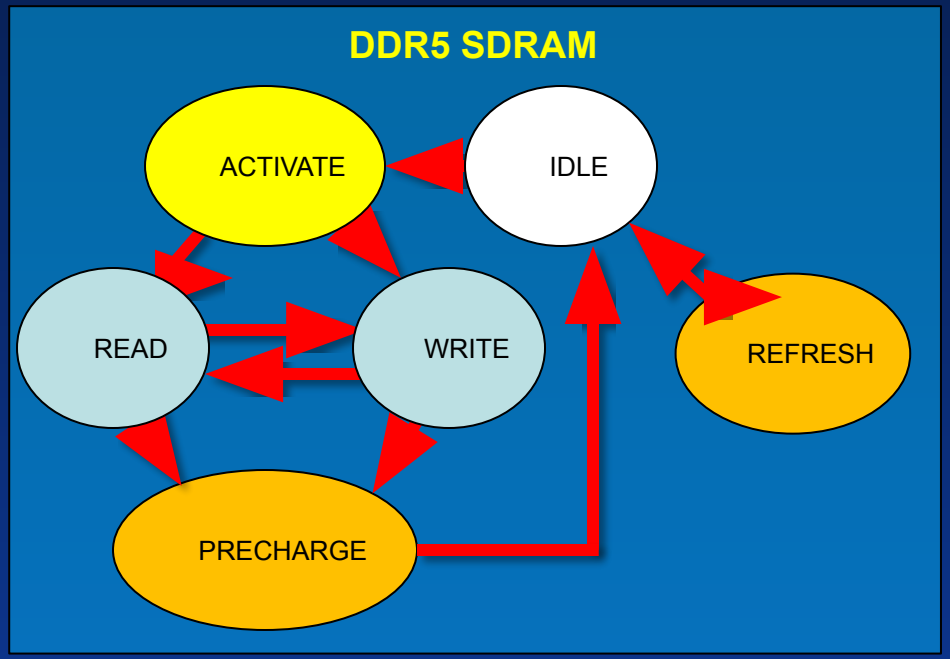


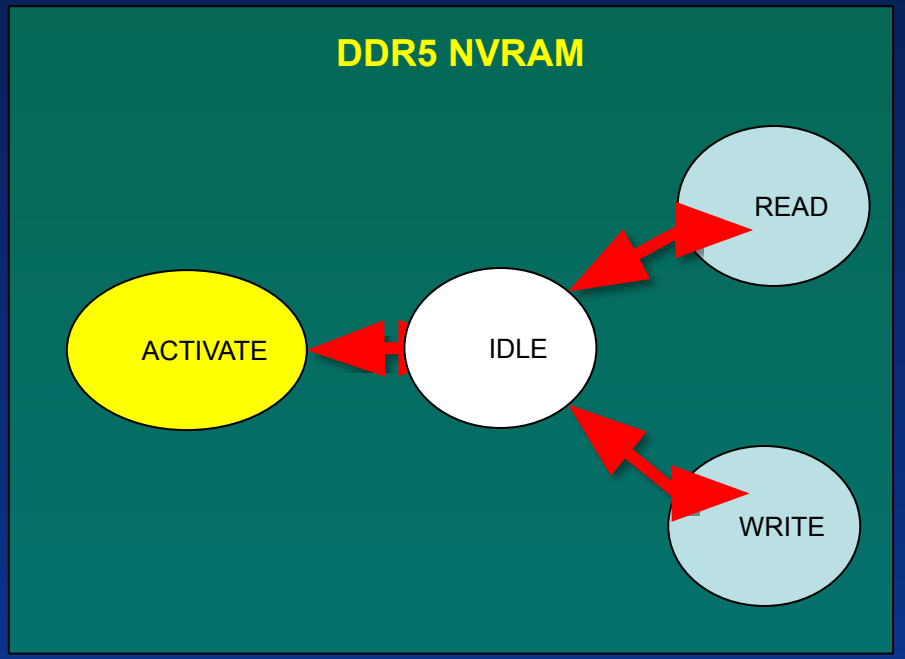
Differences captured in the SPD

DDR5 NVRAM Core Functions

Activate	Fetches data from the array to sense amps
Read	Read data from sense amps
Write	Write data to sense amps
Precharge	NOP
Refresh	NOP
Self Refresh	NOP

Closer to the dream of a LOAD/STORE memory





Compatible but more efficient

NVRAM scales much better than DRAM

16 Gb
28 nm
4 layers CNT

New process

64 Gb
14 nm
4 layers CNT

Add layers

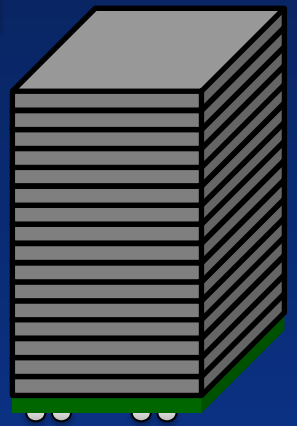
128 Gb
14 nm
8 layers CNT

New process

256 Gb
7 nm
4 layers CNT

Add layers

512 Gb
7 nm
8 layers CNT



16-die stacks

Expect up to 512 Gb/die or 2 TB/package in DDR5 market window

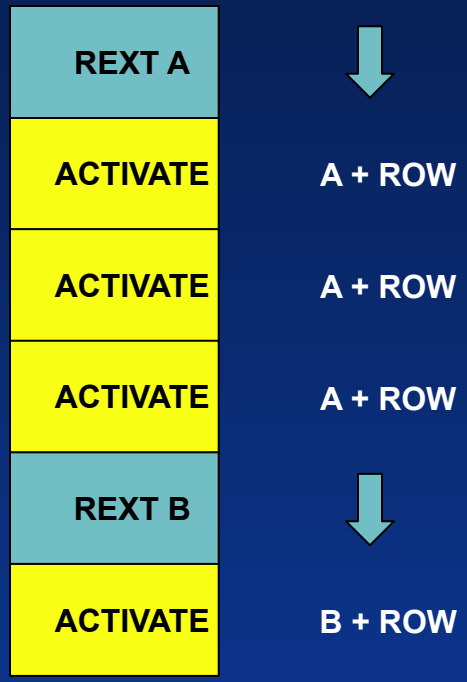
Won't the DDR5 limit of 32Gb per die limit NVRAM?

Introducing Row Extension

Based on the 1980s "Expanded Memory" concept

Adds a sort of paging register

Improved semantics for efficiency by only impacting ACTIVATE



Function	Abbr	CS	CA0	CA1	CA2	CA3	CA4	CA5	CA6	CA7	CA8	CA9	CA10	CA11	CA12	CA13
Row Extension	REXT	L	L	H	R17	R18	R19	R20	R21	R22	R23	R24	R25/B8	R26/B7	R27/B6	R28/B5

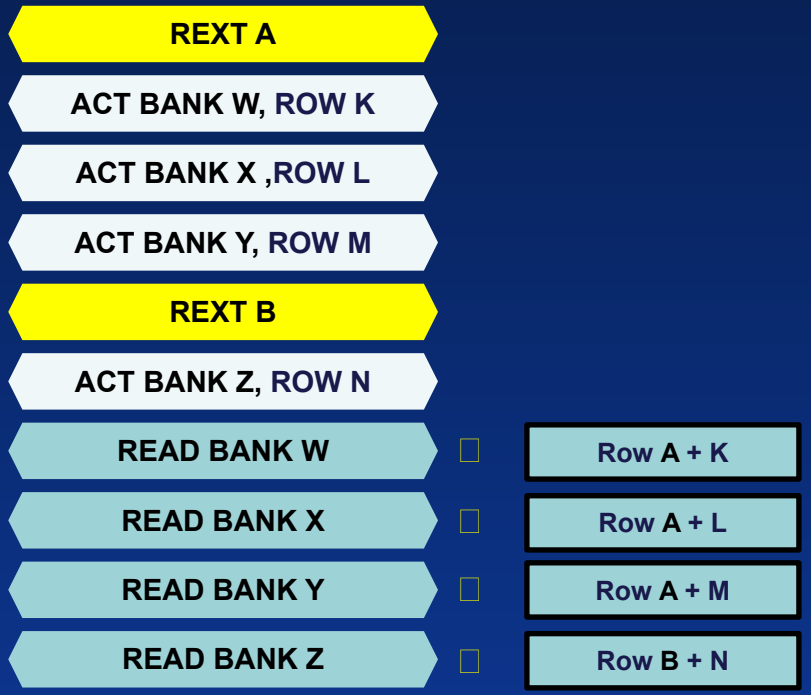
New command

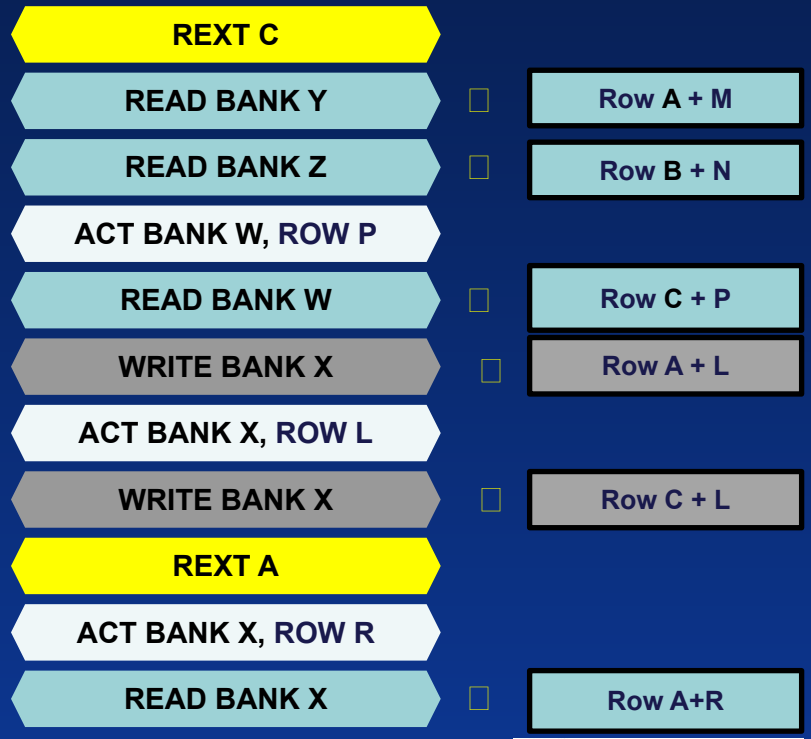
Row Extension

Row/Bank

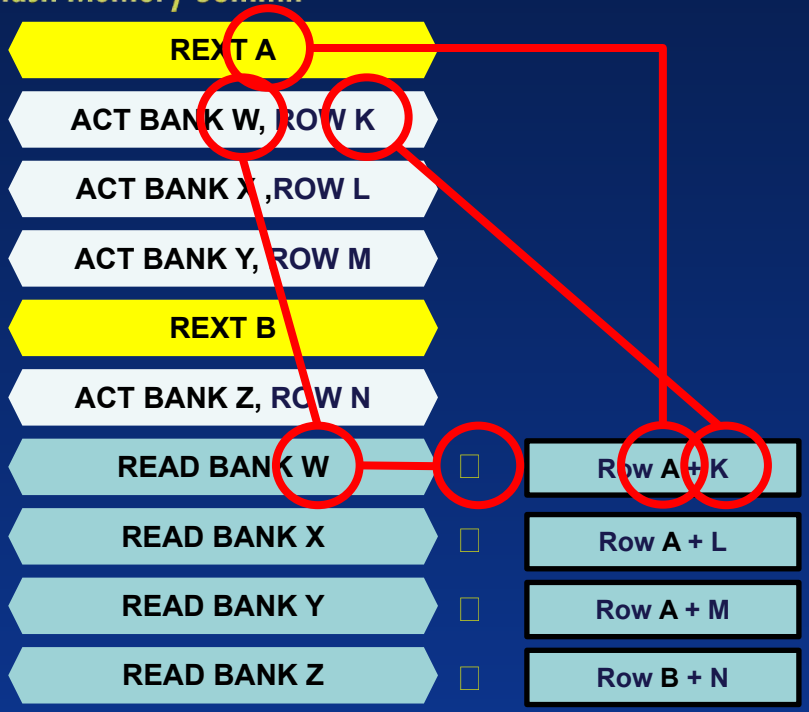
COLUMN

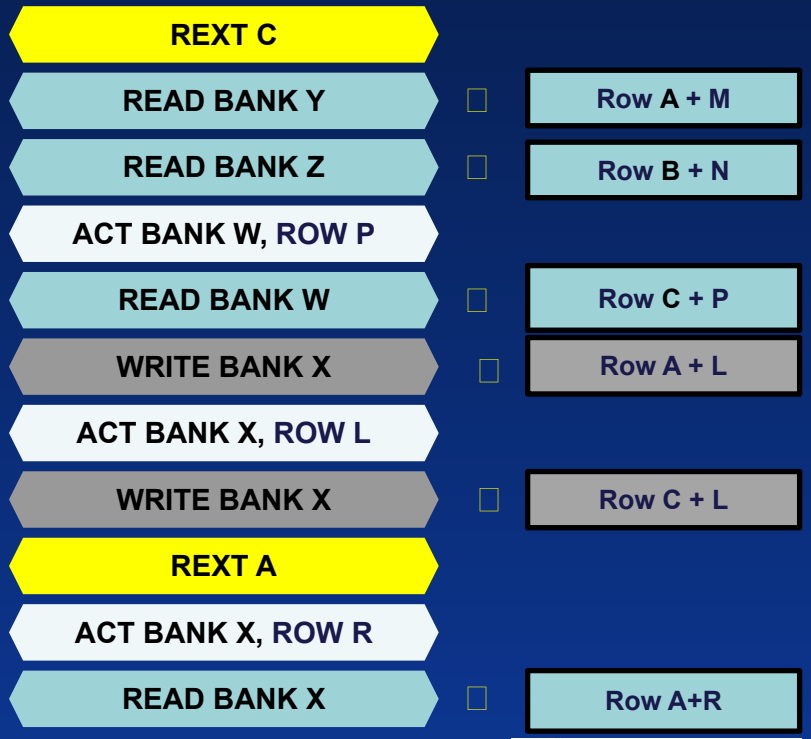
Impact: Extends # address bits associated with bank buffer


REXT A
ACT BANK W, ROW K
ACT BANK X ,ROW L
ACT BANK Y, ROW M
REXT B
ACT BANK Z, ROW N
READ BANK W
Row A + K
READ BANK X
Row A + L
READ BANK Y
Row A + M
READ BANK Z
Row B + N

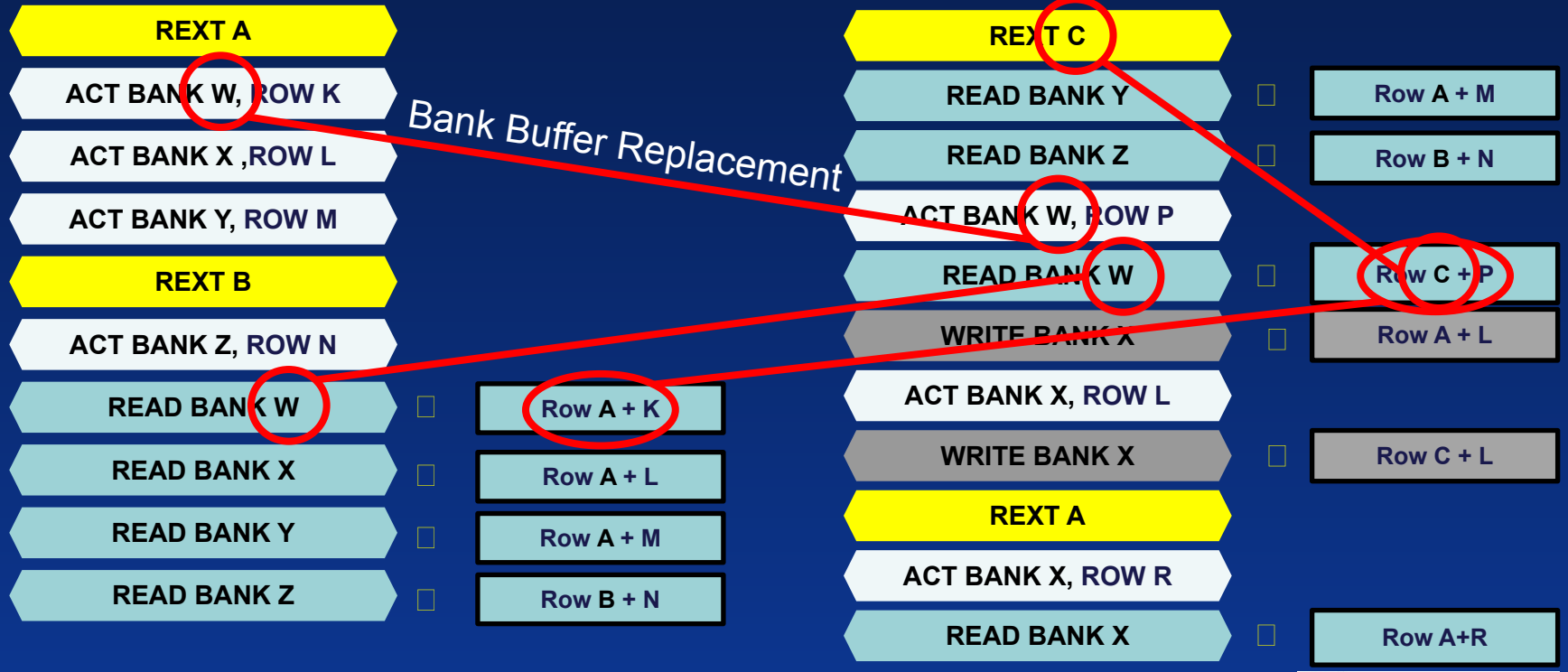

REXT C
READ BANK Y
Row A + M
READ BANK Z
Row B + N
ACT BANK W, ROW P
READ BANK W
Row C + P
WRITE BANK X
Row A + L
ACT BANK X, ROW L
WRITE BANK X
Row C + L
REXT A
ACT BANK X, ROW R
READ BANK X
Row A+R

Row Extension Example



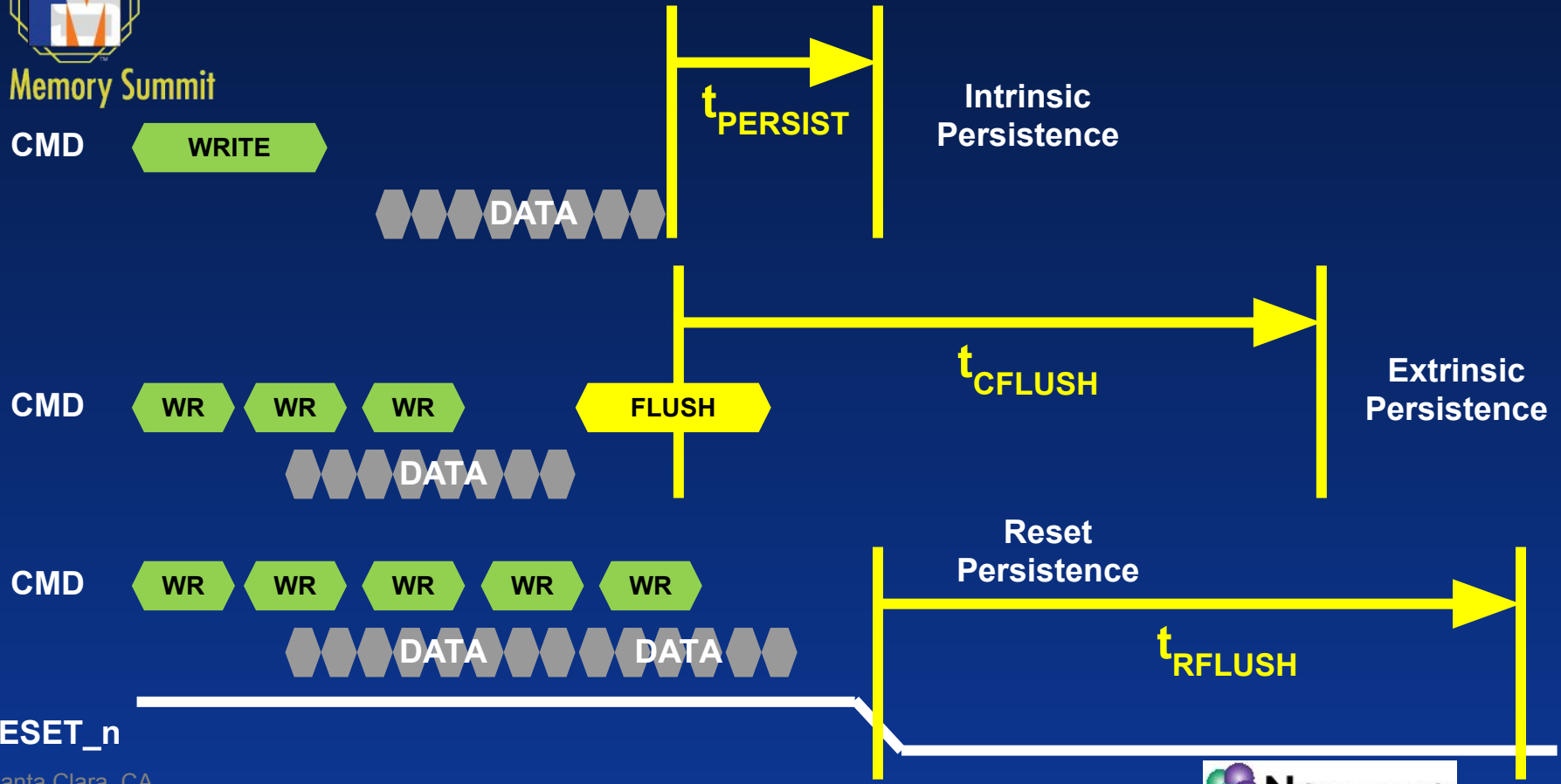


Row Extension Replacement Example



"Power fail isn't the only concern"
"Need to know exactly when data is committed to NVM cells"

New persistence definitions address this


CMD
WRITE
DATA
tPERSIST
Intrinsic Persistence
CMD
WR
WR
WR
FLUSH
DATA
tCFLUSH
Extrinsic Persistence
CMD
WR
WR
WR
WR
WR
DATA
DATA
Reset Persistence
tRFLUSH
RESET_n

JEDEC®

Provides a safe place for competitors to jointly develop specifications

Wider, more rapid acceptance

Documents features for controller designers

Summary

- Saving Data is a Pain
- Need tiers of memory & storage
- Power Fail Sucks
- Persistence is Essential
- NVRAM Spec Allows Many Types
- DDR5 NVRAM Spec in Progress

Questions?

Bill Gervasi
Principal Systems Architect
bilge@Nantero.com