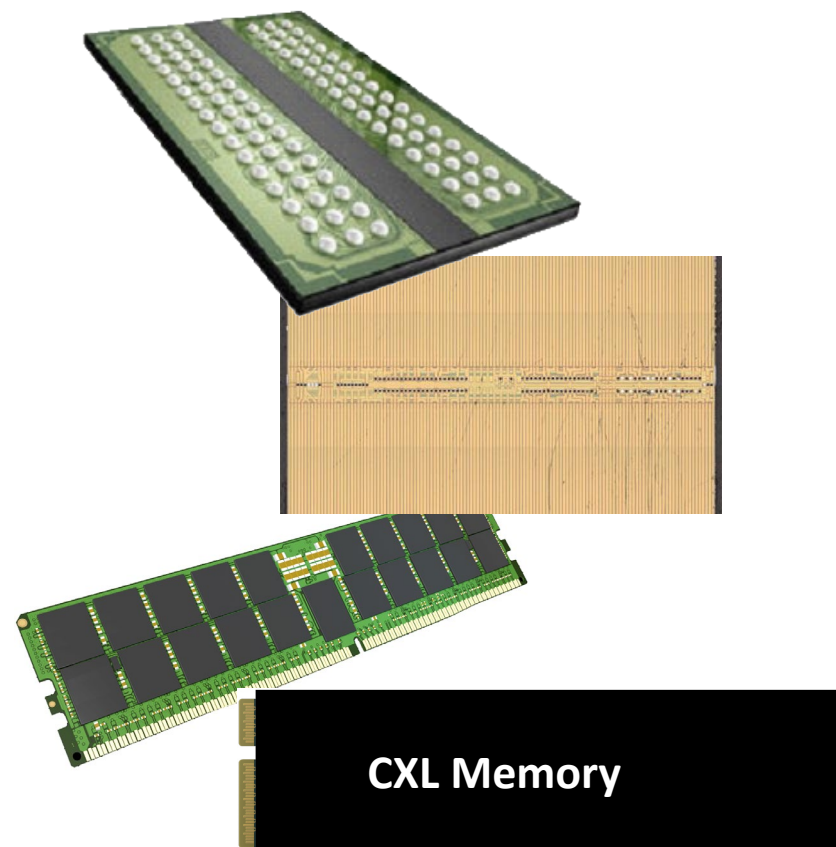


Introduction to DRAM Technology

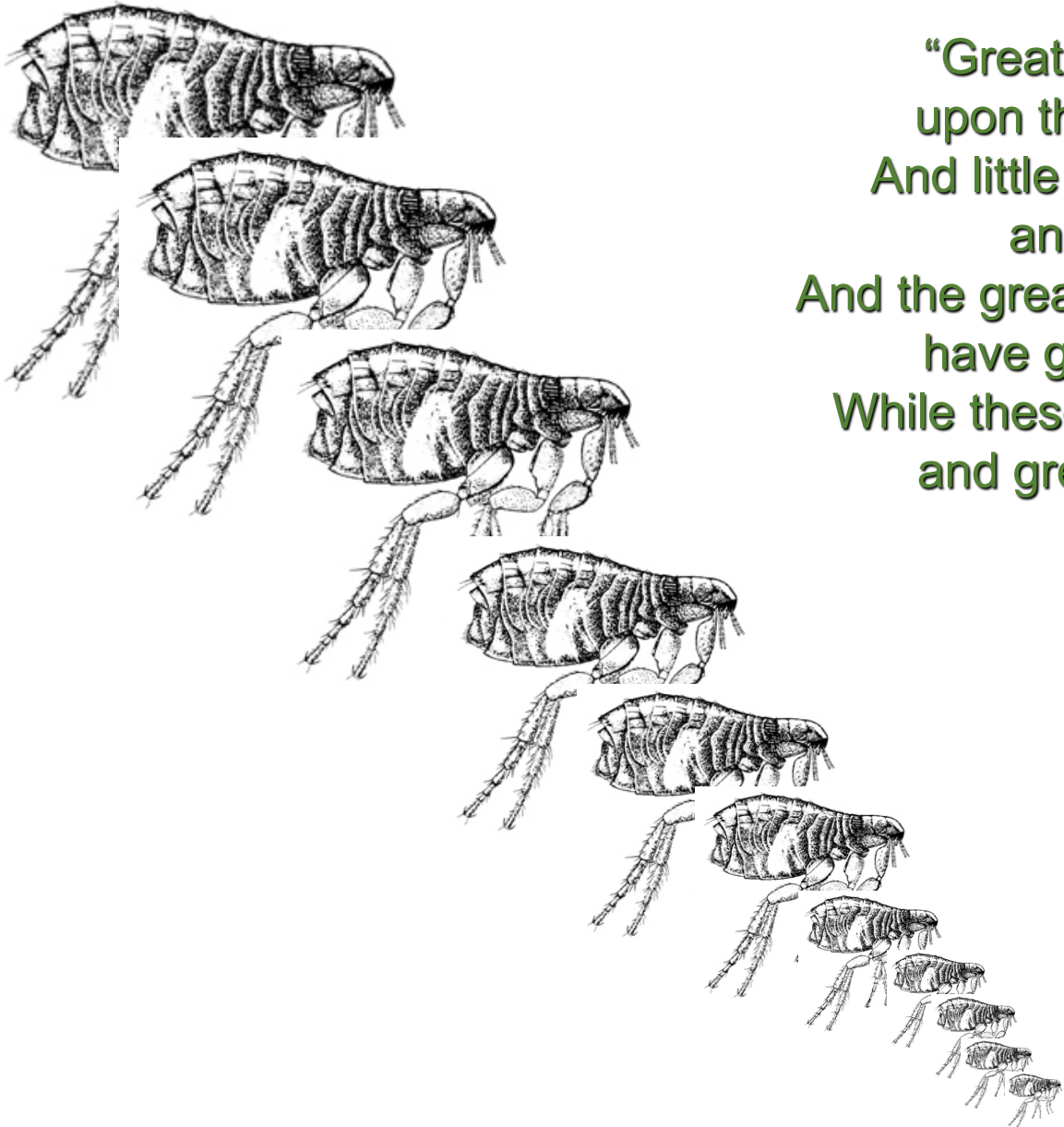
Bill Gervasi, Principal Systems Architect
Wolley Inc.
bilge@wolleytech.com





How we Hit the Memory Wall

And How We'll Get Over It



“Great fleas have little fleas
upon their backs to bite ’em,
And little fleas have lesser fleas,
and so ad infinitum.
And the great fleas themselves, in turn,
have greater fleas to go on;
While these again have greater still,
and greater still, and so on.”

Jonathan Swift

DRAM so far has resisted revolution

Just a number of evolutionary changes

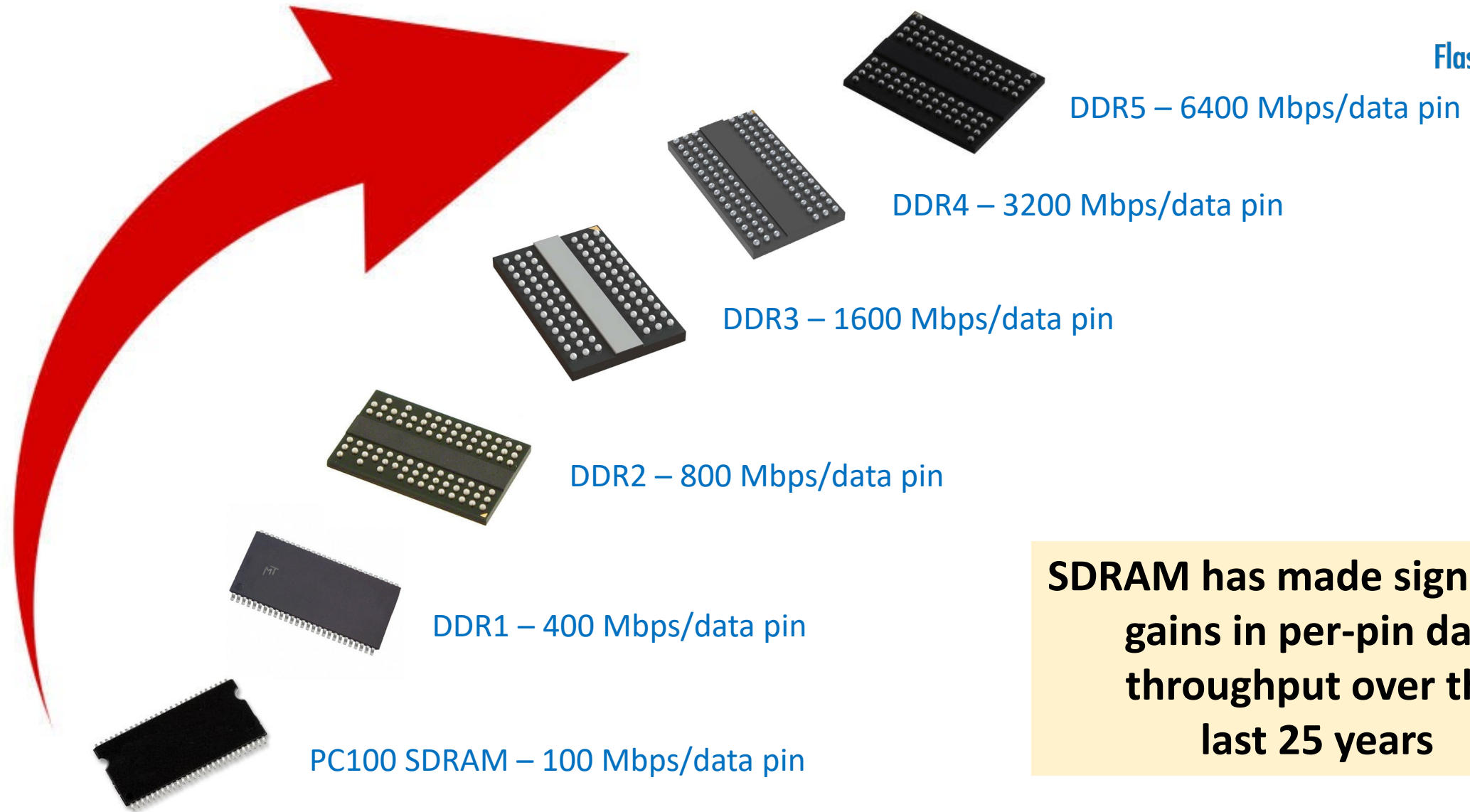
We are still using a core design >300 years old



Why?

Customers pay for GB and not much else matters





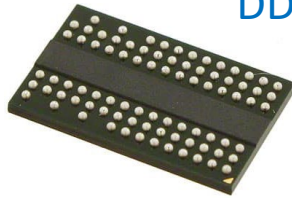
SDRAM has made significant gains in per-pin data throughput over the last 25 years



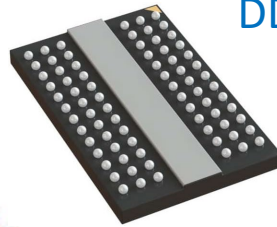
PC100 SDRAM – reference synchronous main memory



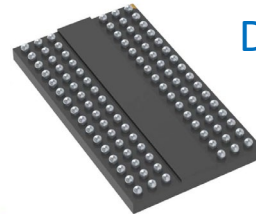
DDR1 – prefetch 2 bits, first main memory with a data strobe



DDR2 – prefetch 4 bits, differential strobes, on-die termination



DDR3 – prefetch 8 bits, improved calibration, command-dependent ODT



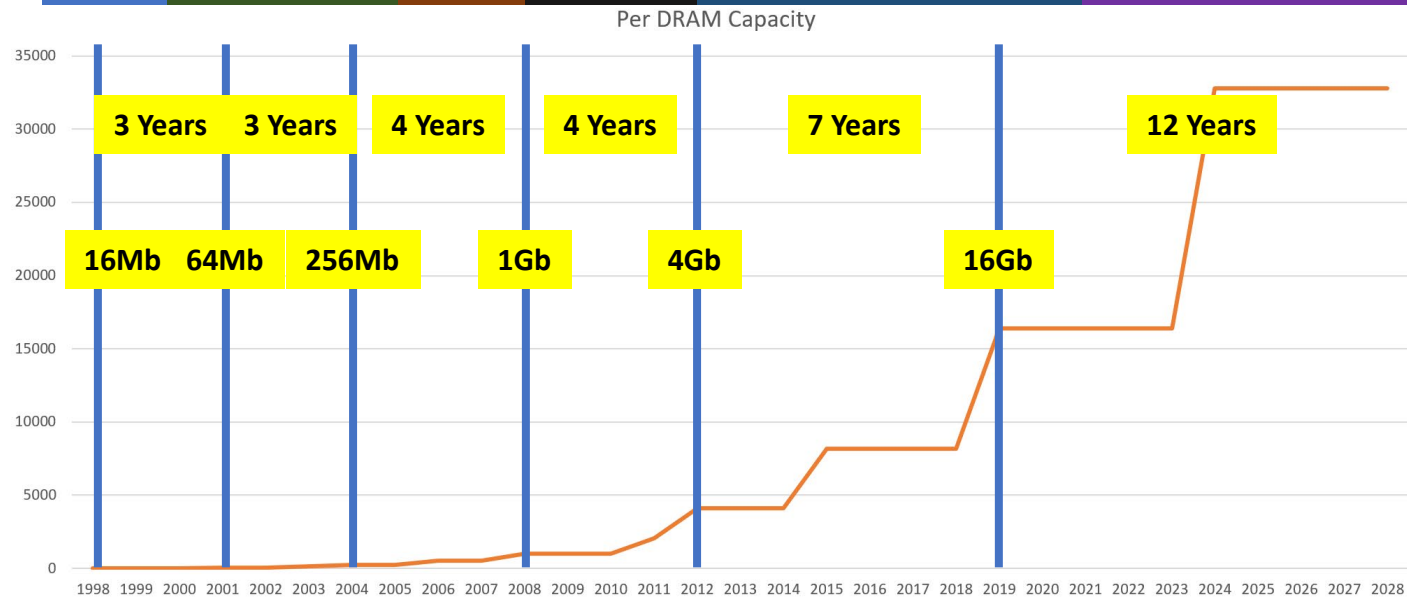
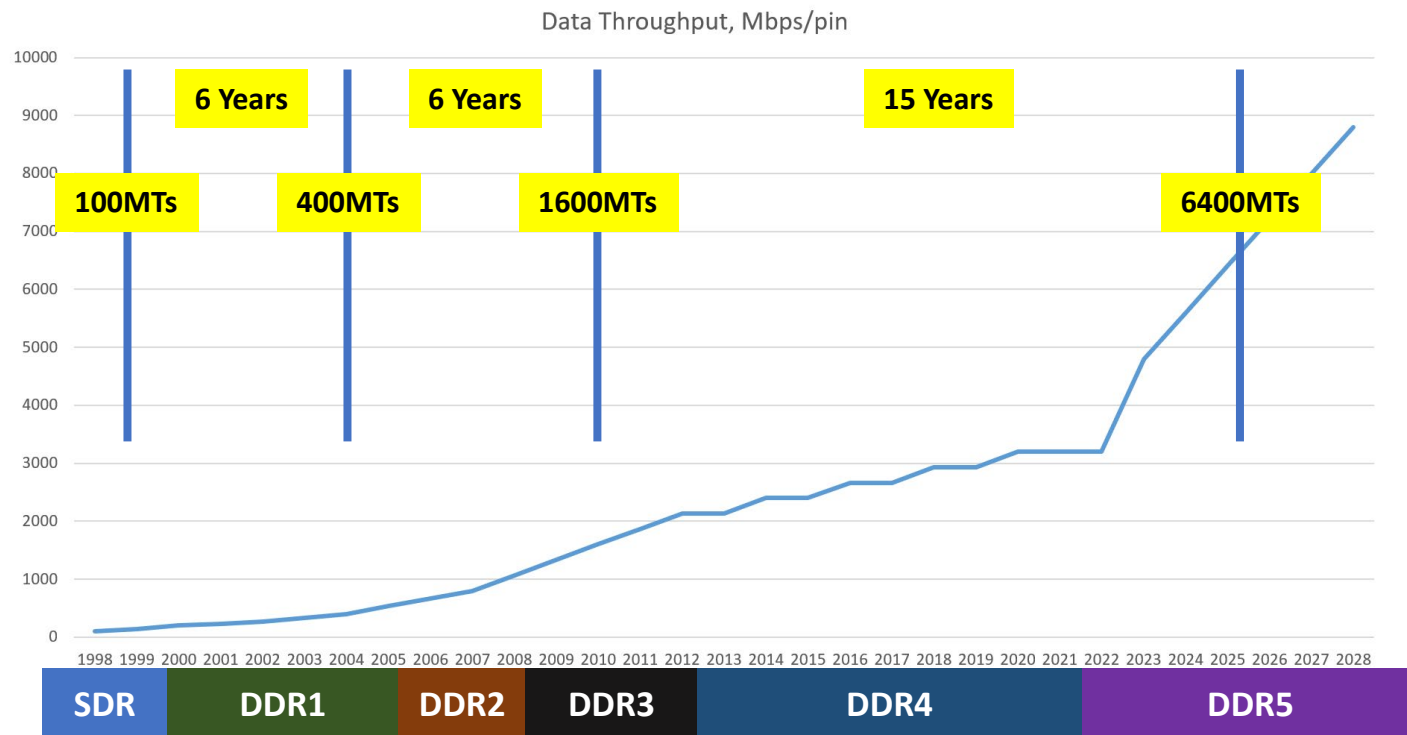
DDR4 – improved calibration, ODT



DDR5 – Prefetch 16, improved calibration, PMIC

**However,
random access time
has only improved
28%**

**‘cuz I/O is cheaper
than core**



The good news:

Data throughput has had healthy increases

DDR5 was planned for 6400 Mbps max,
now extended to 8800 Mbps

The bad news:

Speed improvements slowing

DRAM per-die capacity is taking longer
with each generation

Was: quadrupling every 3 years

Is: quadrupling every 12 years



How do these trends
affect my system design?

How do I make the most
of what we have?



DDR5 SDRAM

JESD79-5

Published: Jul 2020

This document defines the DDR5 SDRAM specification, including features, functionalities, AC and DC characteristics, packages, and ball/signal assignments. The purpose of this Standard is to define the minimum set of requirements for JEDEC compliant 8Gb through 32Gb for x4, x8, and x16 DDR5 SDRAM devices. This standard was created based on the DDR4 standards (JESD79-4) and some aspects of the DDR, DDR2, DDR3 & LPDDR4 standards (JESD79, JESD79-2, JESD79-3 & JESD209-4). Item 1848.99G.

Committee(s): JC-42.3B

Available for purchase: **\$369.00** [Add to Cart](#) [i](#)

Paying JEDEC Members may [login](#) for free access.

Request Assistance

Standards & Documents Assistance:

Email [Julie Carlson](#)

For other assistance, including website or account help, [contact JEDEC by email here](#).

**Except when needed, this tutorial
will focus on DDR5 SDRAM
and related modules**

Available from jedec.org

JEDEC standards and publications are copyrighted by the JEDEC Solid State Technology Association. All rights reserved.



Today's Agenda

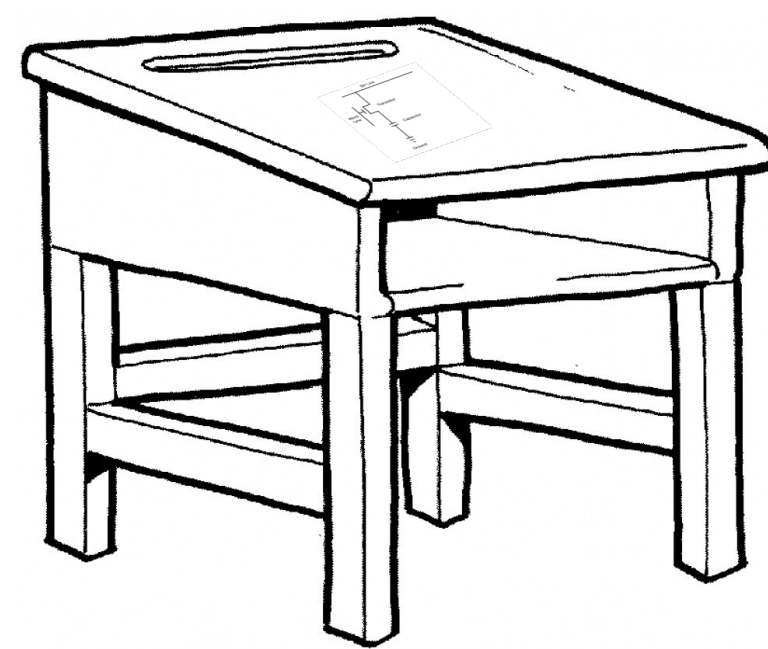
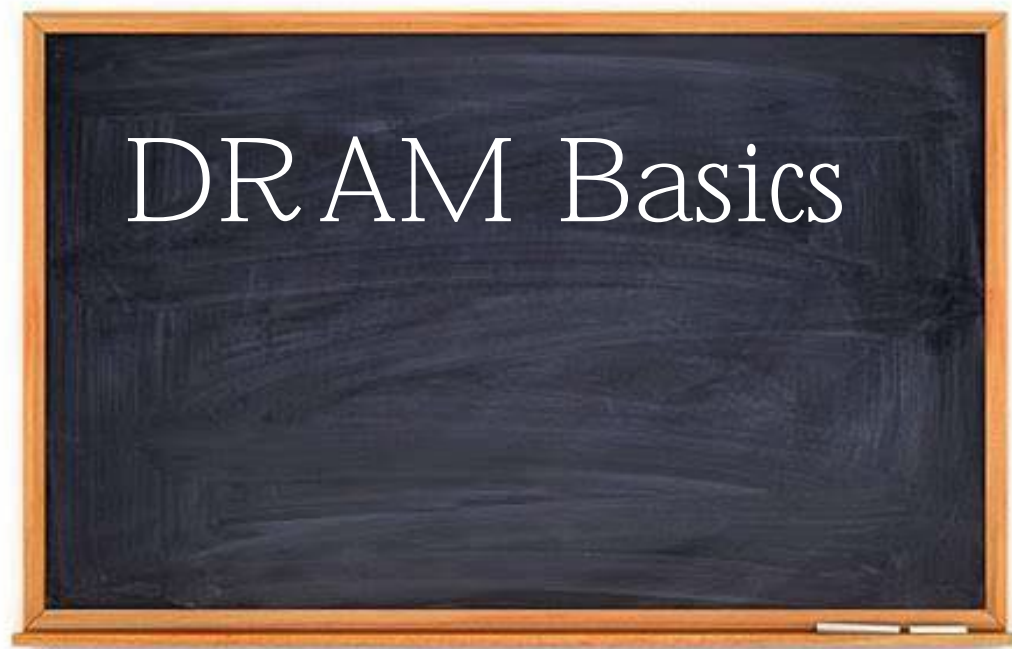
DRAM Basics

DRAM Variations

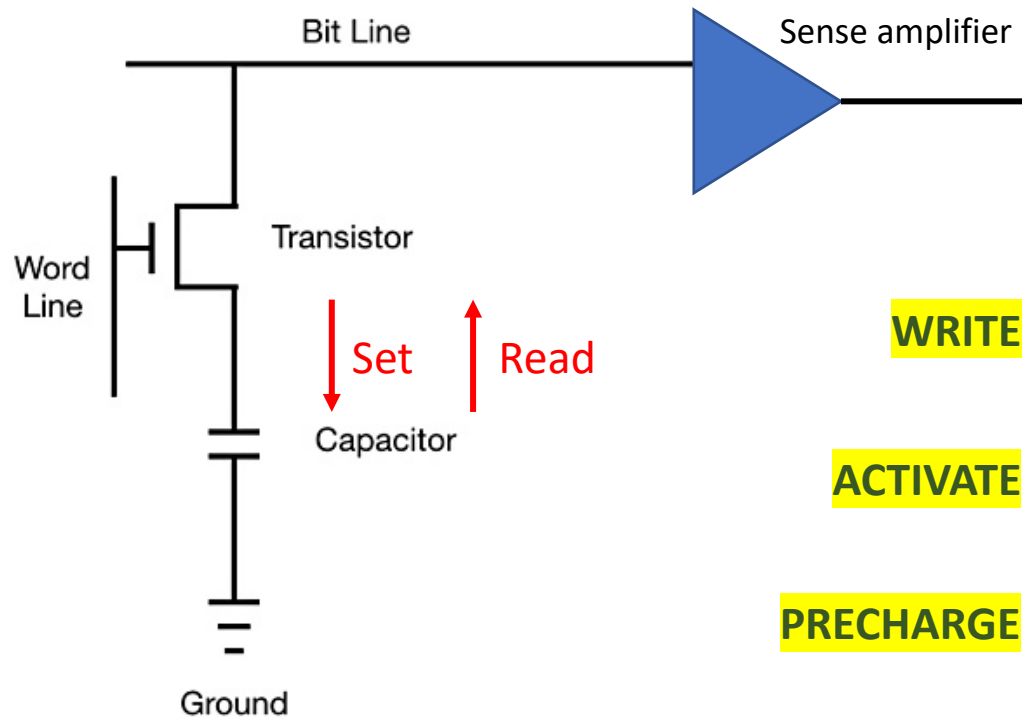
DRAM Memory Modules

New Vectors in Memory: Chiplets & CXL

Memory Related Trends & Issues



The basis of all DRAMs is an itty bitty capacitor used to store a charge that represents a bit of data



Setting data in a DRAM cell requires applying a charge to the capacitor

Reading the DRAM cell destroys the capacitor content

Restoring the contents of the DRAM cell requires rewriting the contents of the bit line back into the cell and setting the bit line to a midpoint voltage



JEDEC defines the minimum specification of “**what**” a DRAM looks like from the outside

JEDEC does not define “**how**” the device meets the specification

Each supplier is free to apply their own magic to meet the standard

Suppliers can implement supersets of the JEDEC standard as well

6F2 (3F x 2f) cell

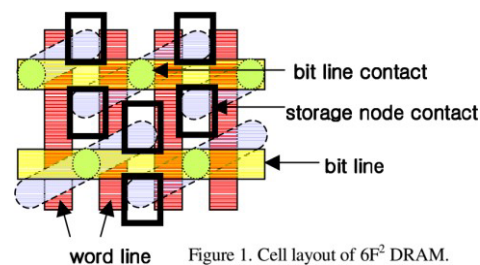
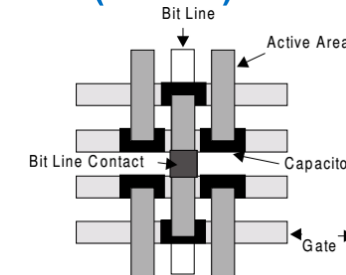


Figure 1. Cell layout of 6F² DRAM.

C. Cho, et. al., “A 6F₂ DRAM Technology in 60nm era for Gigabit Densities,” VLSI’05

8F2 (4F x 2f) cell



S. Bukofsky, et. al., “Extending KrF Lithography to 0.13 μm sub-8F₂ DRAM Technology: The Importance of Lithography-Centric Design,” ESSDERC’00

DRAM Cell Cross-section

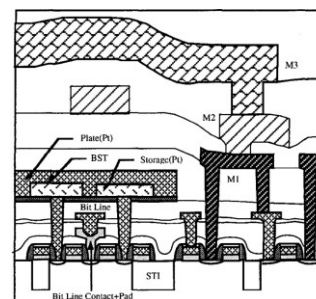
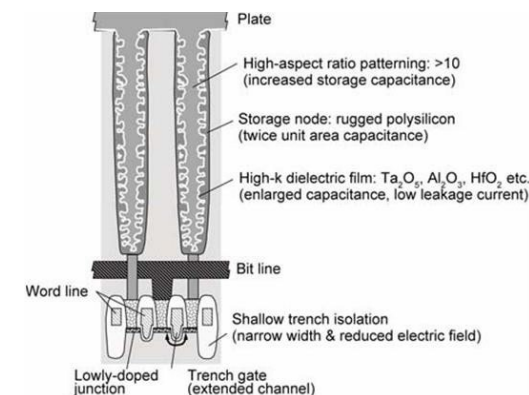
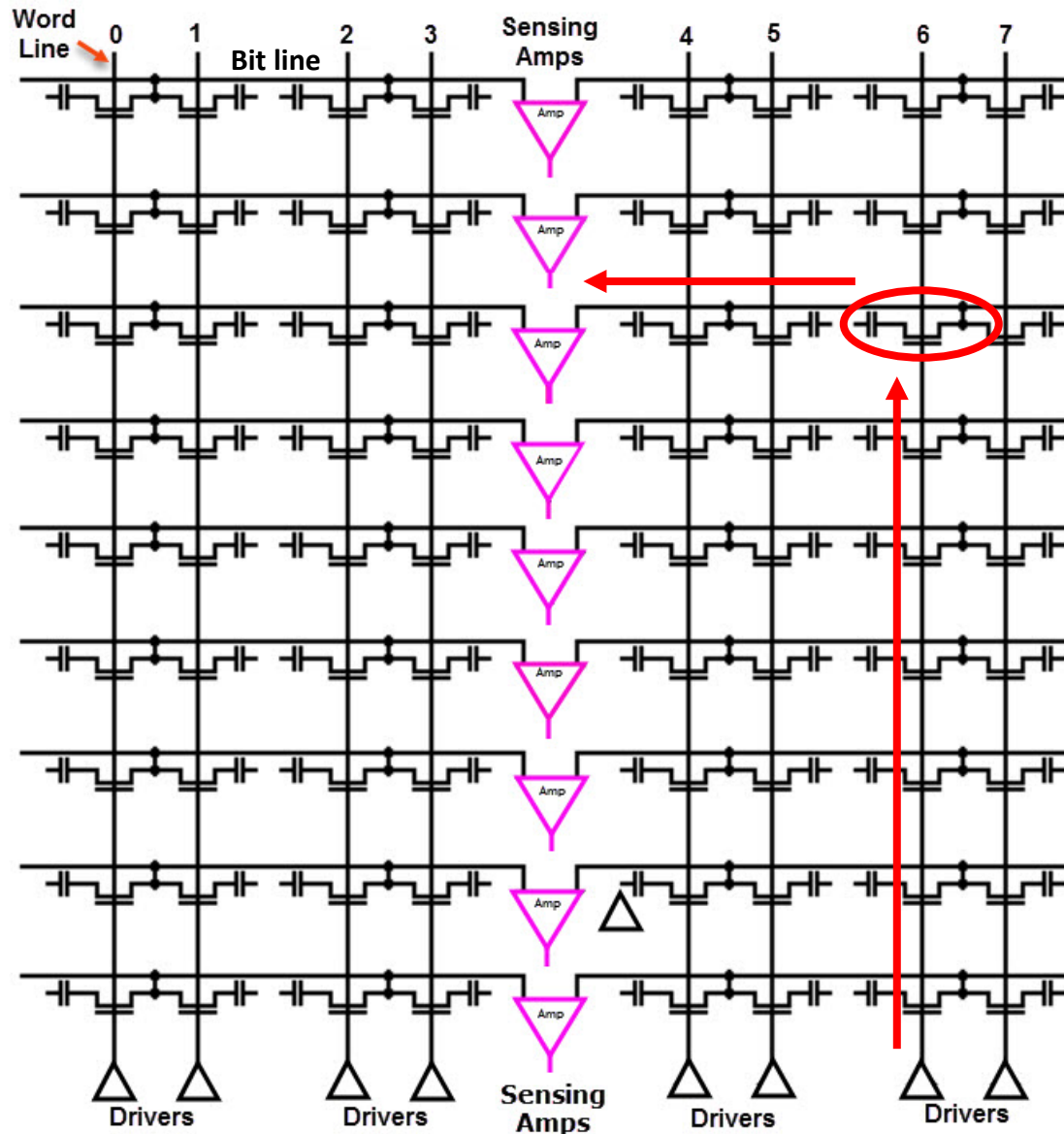


Fig.1 Schematic cross sectional view of the DRAM

K.P. Lee, et. al., “A process technology for 1 giga-bit DRAM,” IEDM’95



H. Sunami, “Dimension Increase in Metal-Oxide-Semiconductor Memories and Transistors”, Advances in Solid State Circuit Technologies Book 2010 Ch15



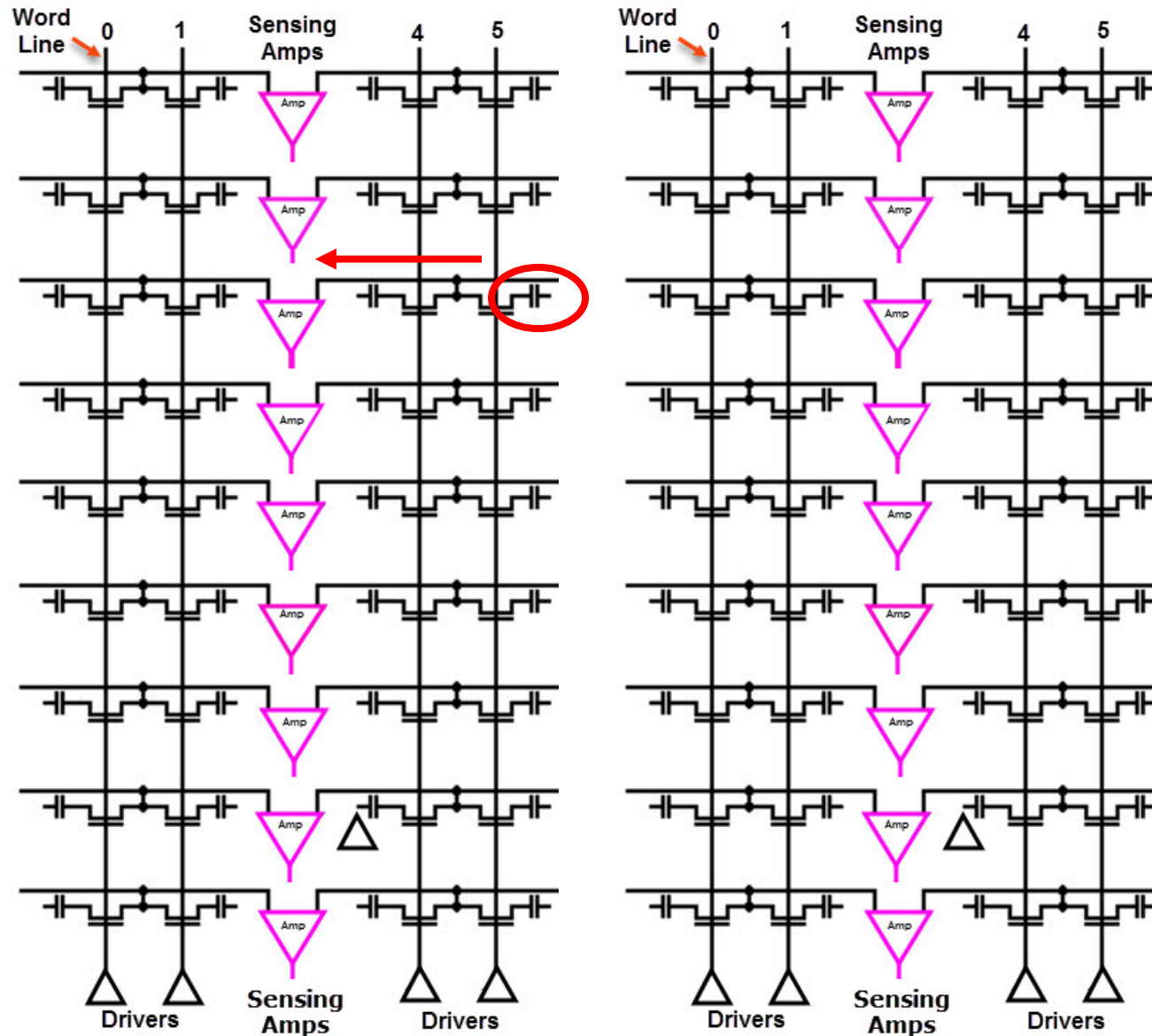
Cells are arranged in a 2-dimensional array with shared word lines and bit lines for many cells

Read access time is largely a function of

- Flight time down the word line +
- Cell discharge +
- Flight time down the bit line
- Sense amplifier throughput

Note the large number of parallel metal lines which are susceptible to crosstalk
...more on this later...

(less obvious but related: shared silicon substrate where cell bleed can occur)



Can DRAM Access Time be Improved?

Since the flight times on word and bit lines are a significant part of access time, why can't we reduce access time?

It's easy to see how access time can be improved by shortening the lines and increasing the number of sense amps – RLDram did this –

however, that increases DRAM cost

Increasing DRAM cost violates the Prime Directive:

“I want it better but at the same price”

Keep it simple – just write and read data

Don't over-engineer

Only add enough features to reach the next level

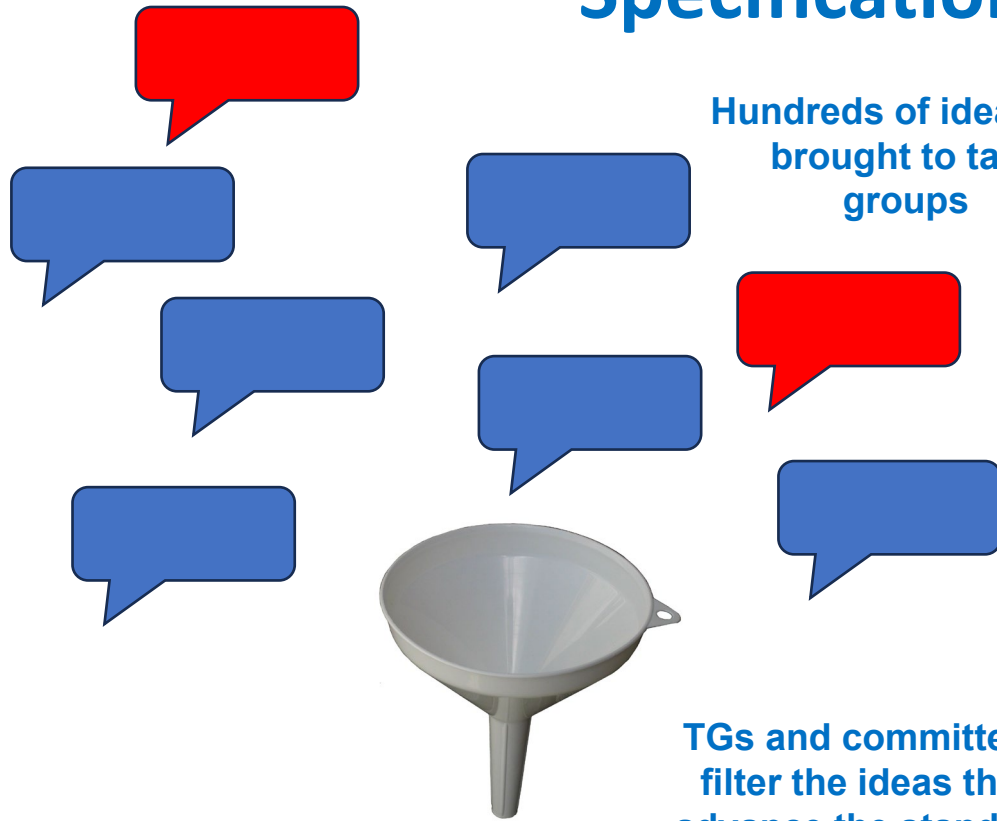


Specification development process

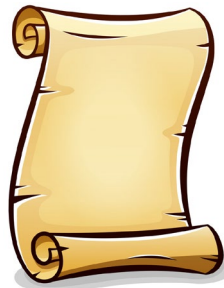


Flash Memory Summit

Hundreds of ideas are brought to task groups



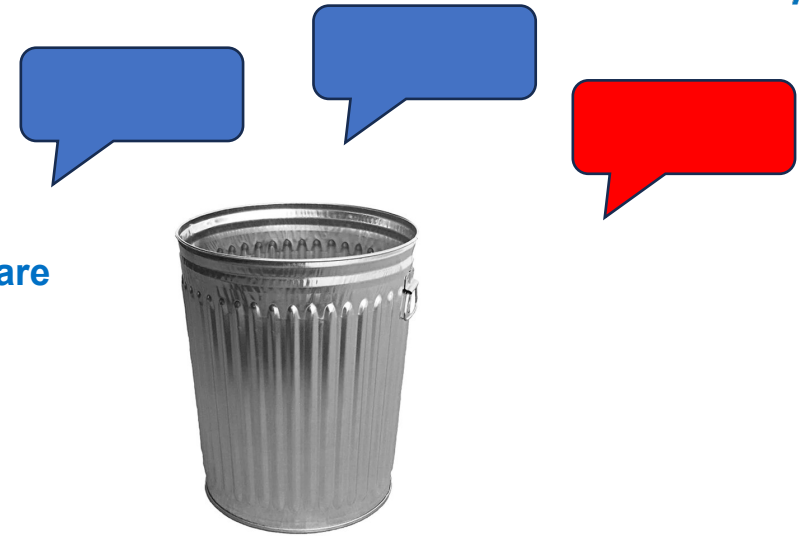
TGs and committees filter the ideas that advance the standard



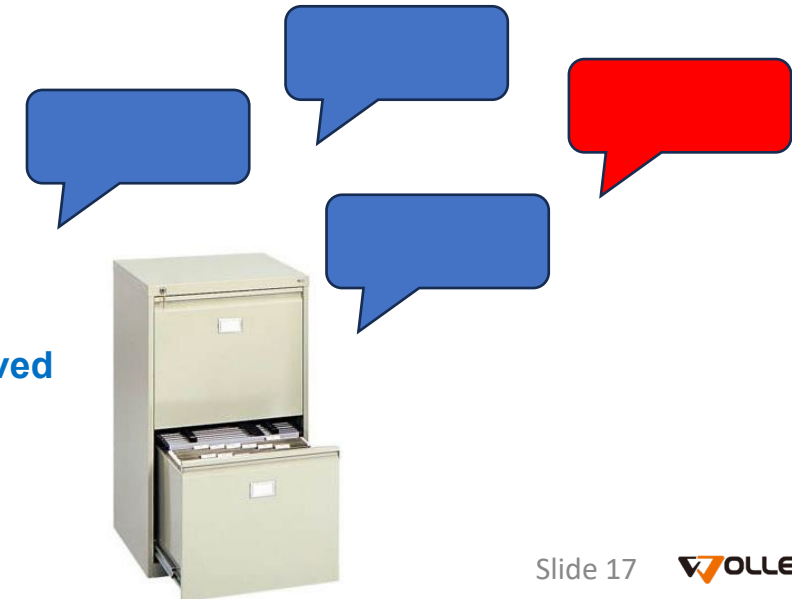
Final standard approved

Standard V2, V3 approved

Some ideas are discarded

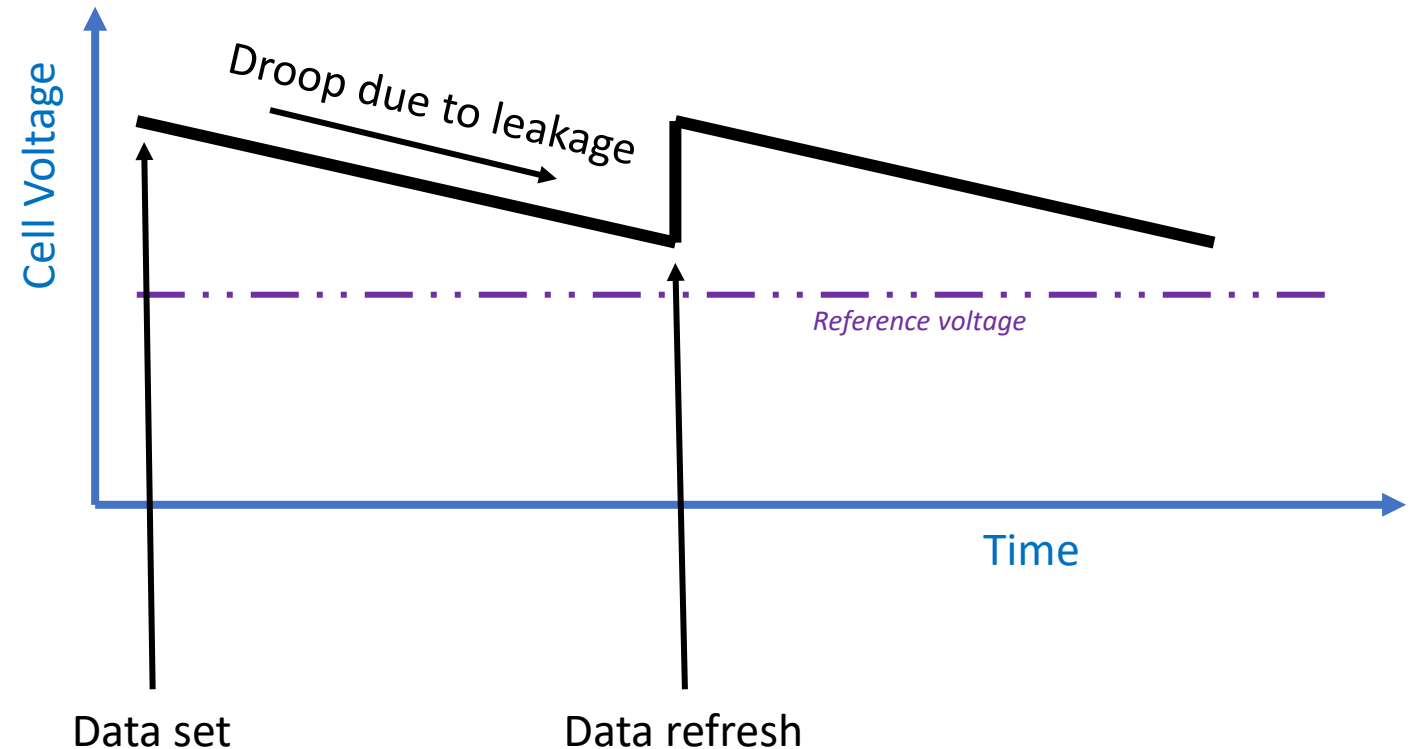
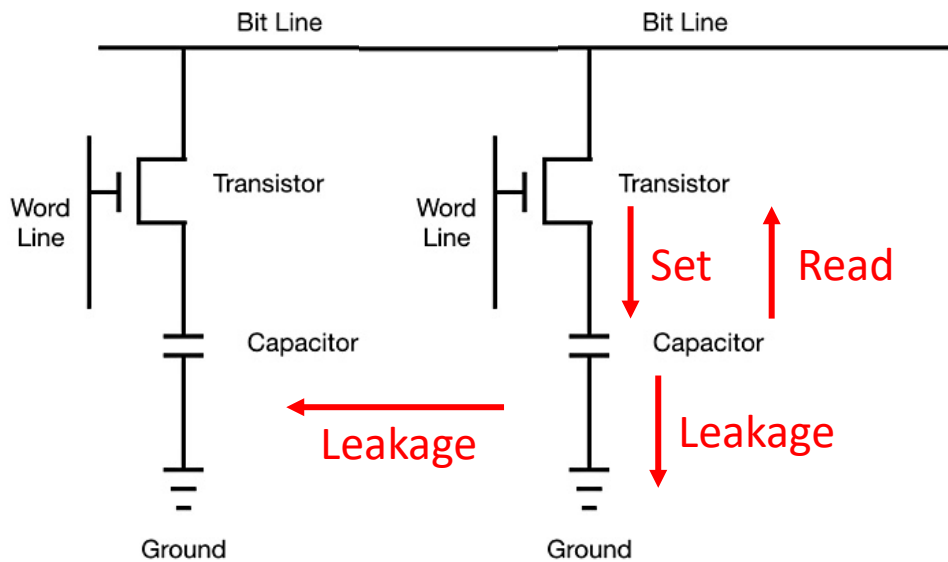


Some ideas are saved for the future



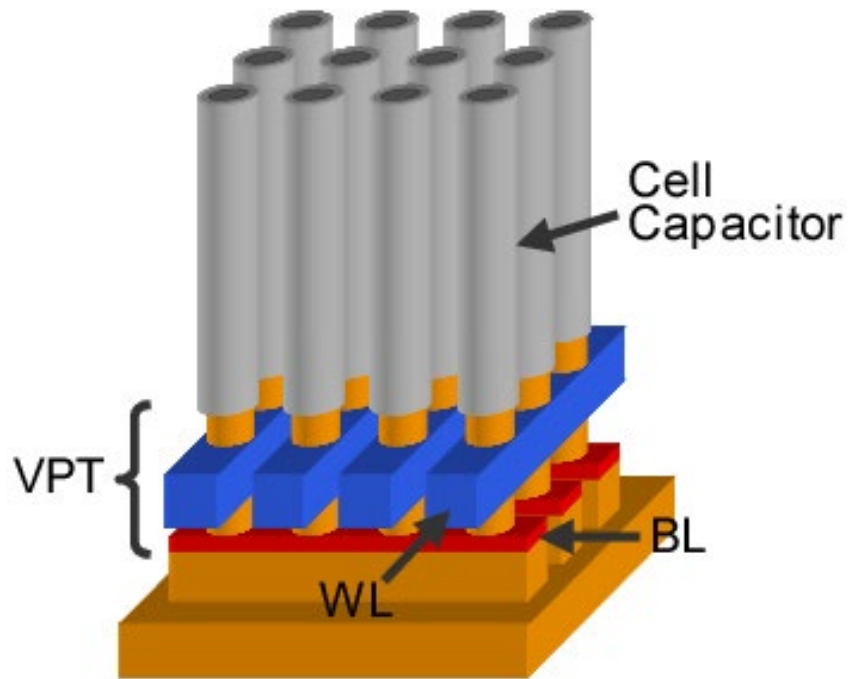
That itty bitty capacitor suffers from leakage (even worse at high temperature)

The D in SDRAM stands for “Dynamic”... over time, the data goes away... requiring a “**Refresh**” operation to restore the cell voltage

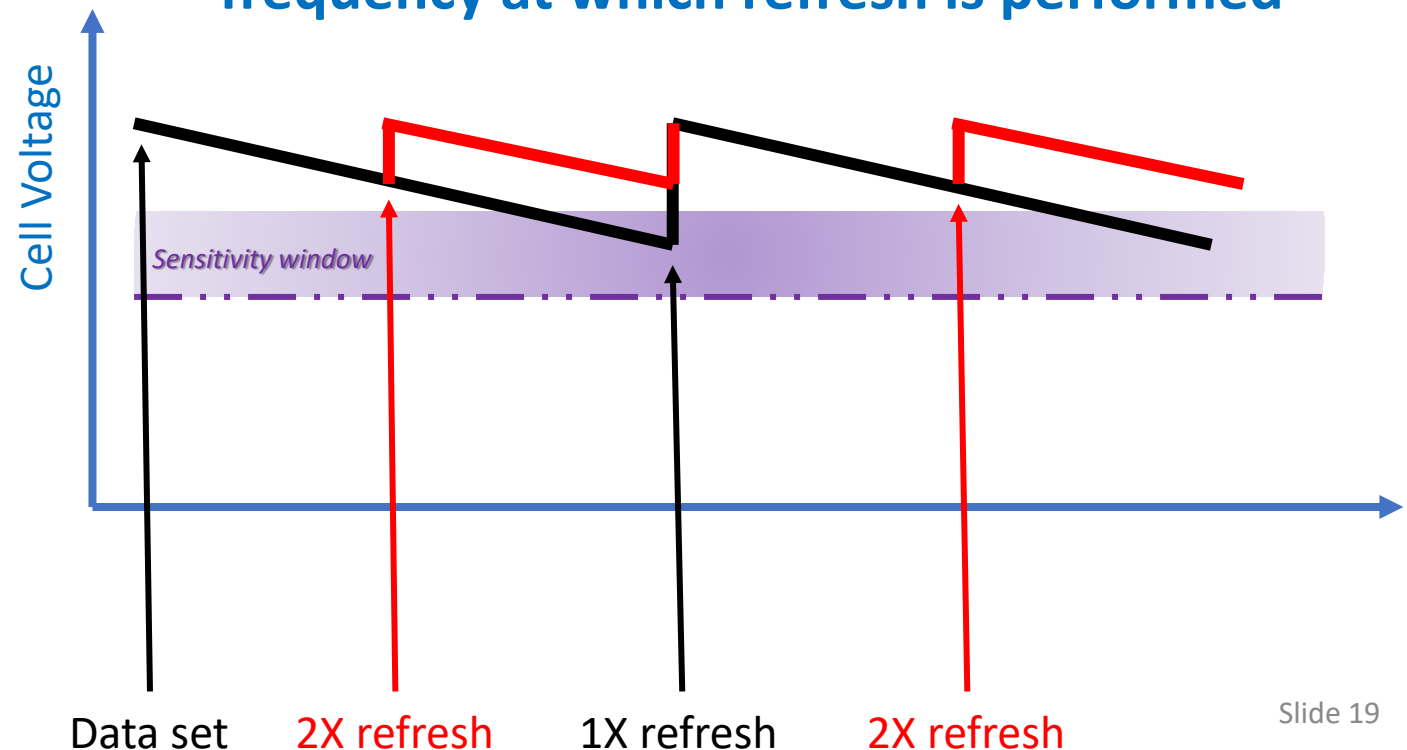
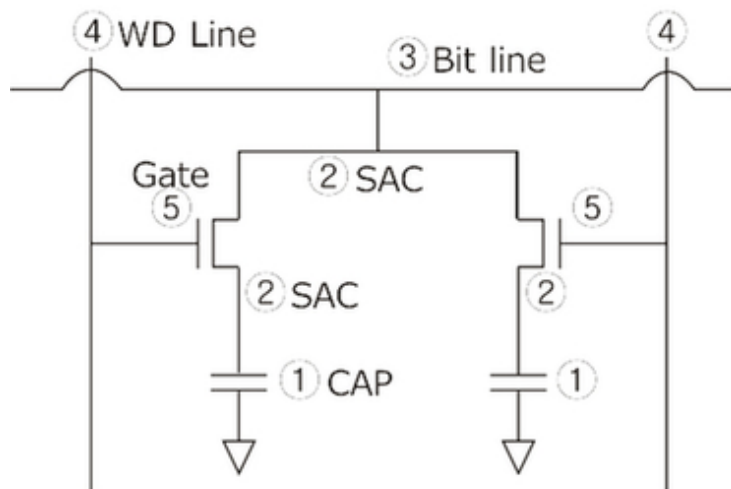




Note that there are a number of paths for cell disruption: crosstalk on word and bit lines as well as between the cell capacitors



Cells become more susceptible as the cell voltage droops – one fix is to increase the frequency at which refresh is performed





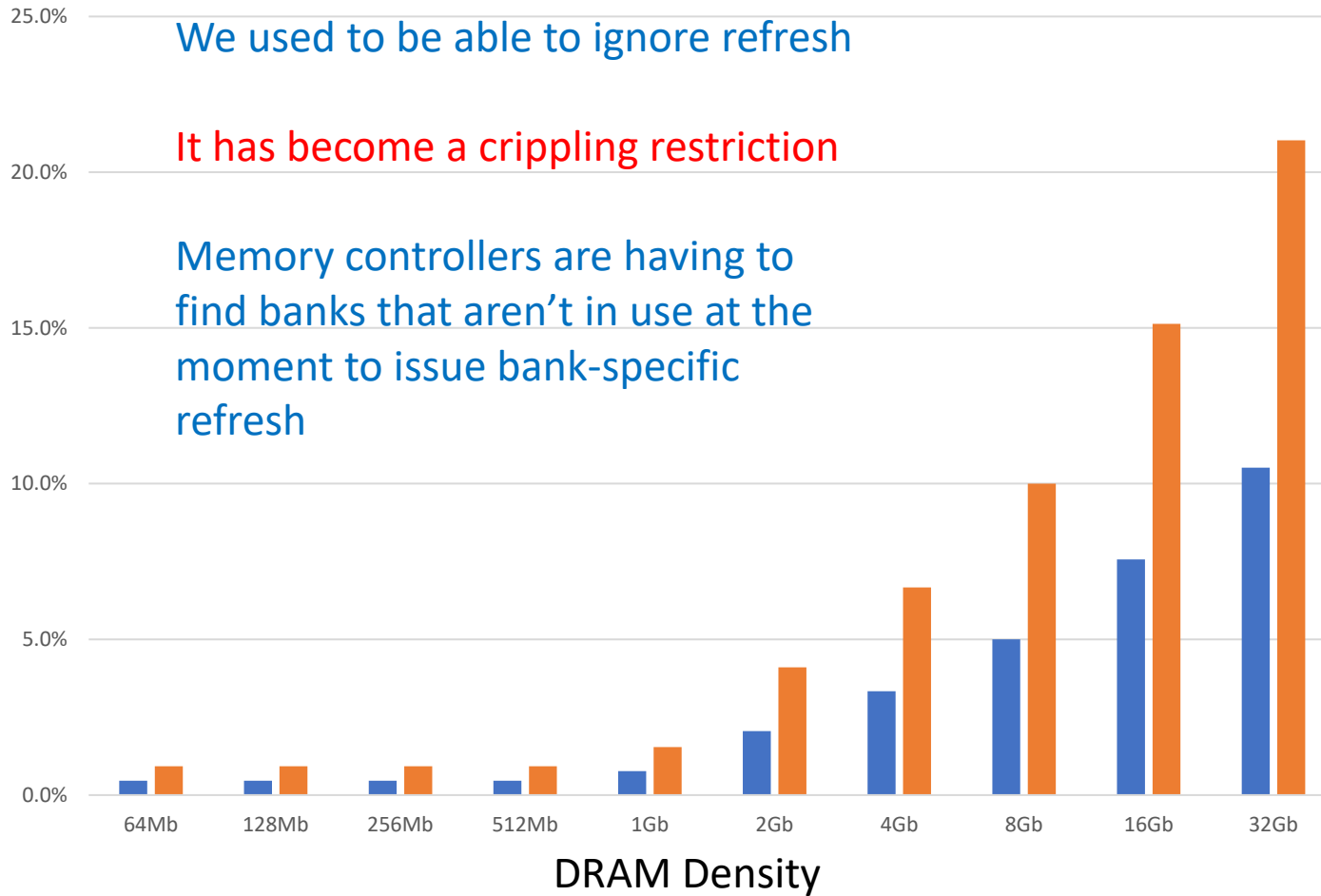
Refresh Penalties, Percentage of Bandwidth

We used to be able to ignore refresh

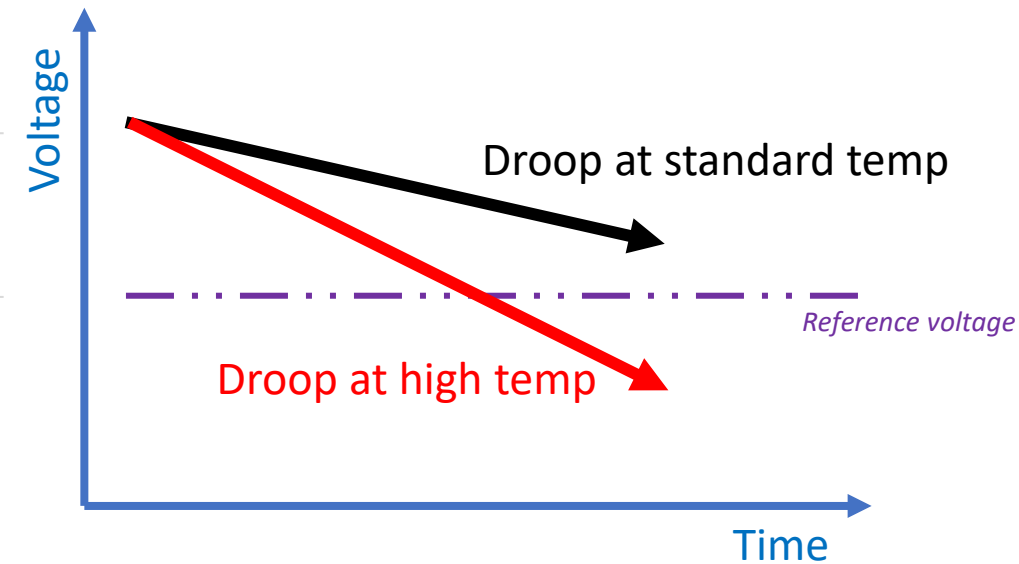
It has become a crippling restriction

Memory controllers are having to find banks that aren't in use at the moment to issue bank-specific refresh

Percentage of Bandwidth



At high temperature (85-95 °C), refresh must be DOUBLED, i.e., **11-21%** of available bandwidth (!!!)



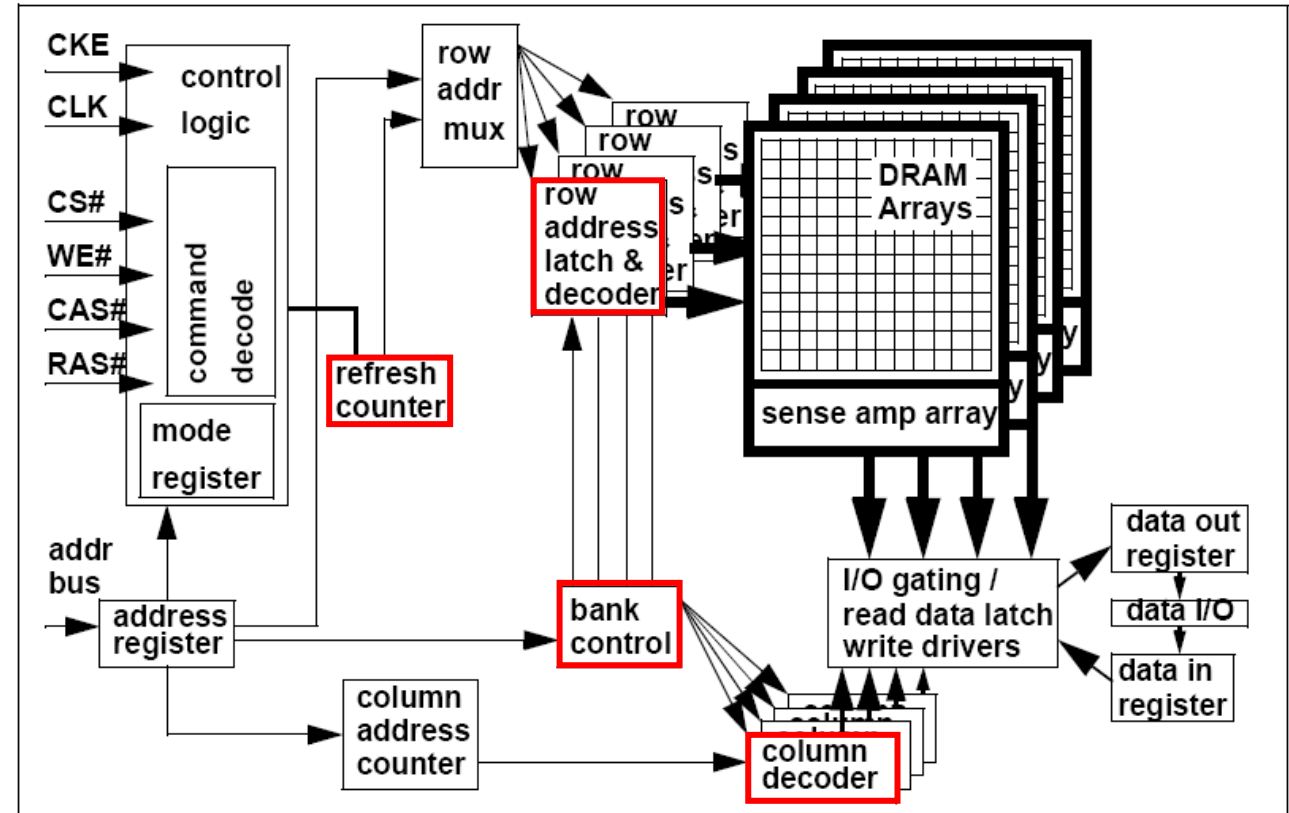


Balancing DRAM array layout and performance, multiple banks are created, each with a row buffer which acts as a cache into the array

At this point, addressing a particular memory location requires:

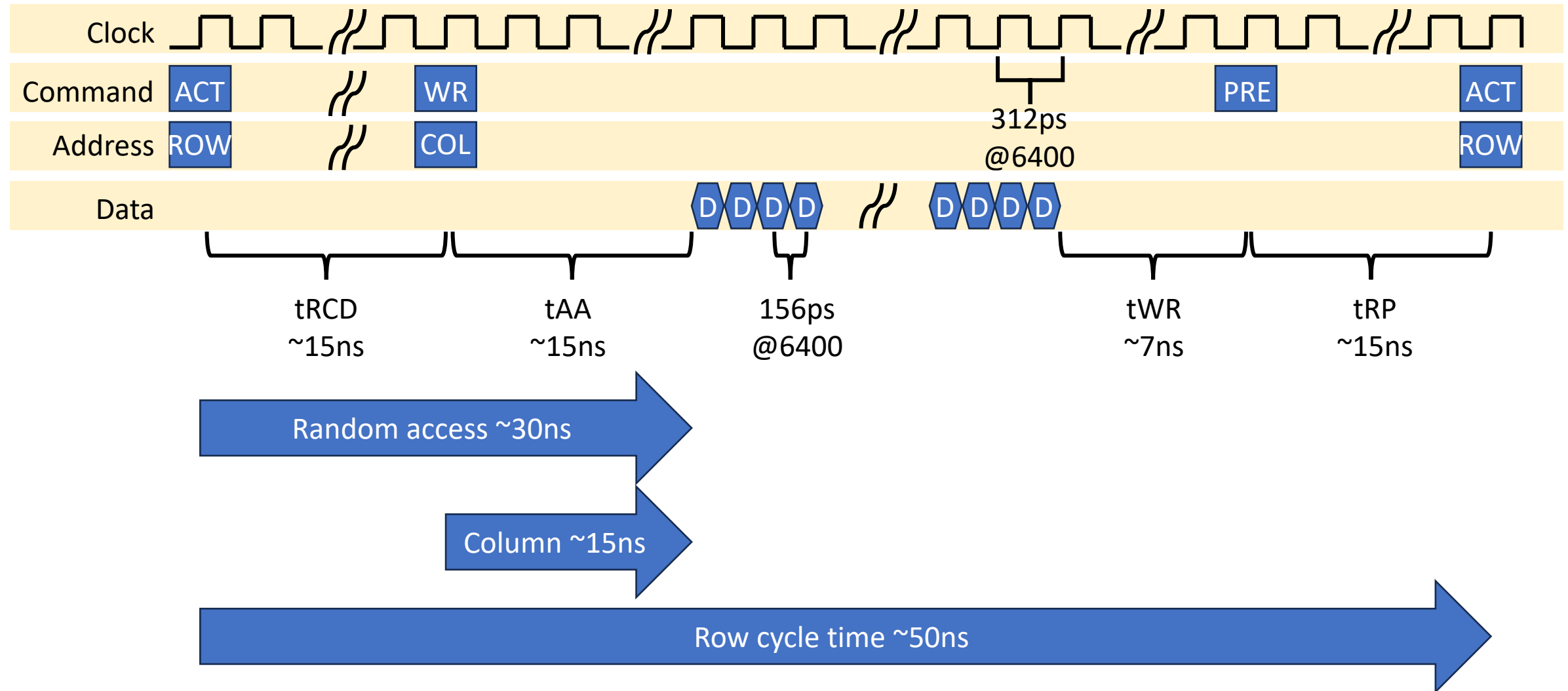
Bank
Row
Column

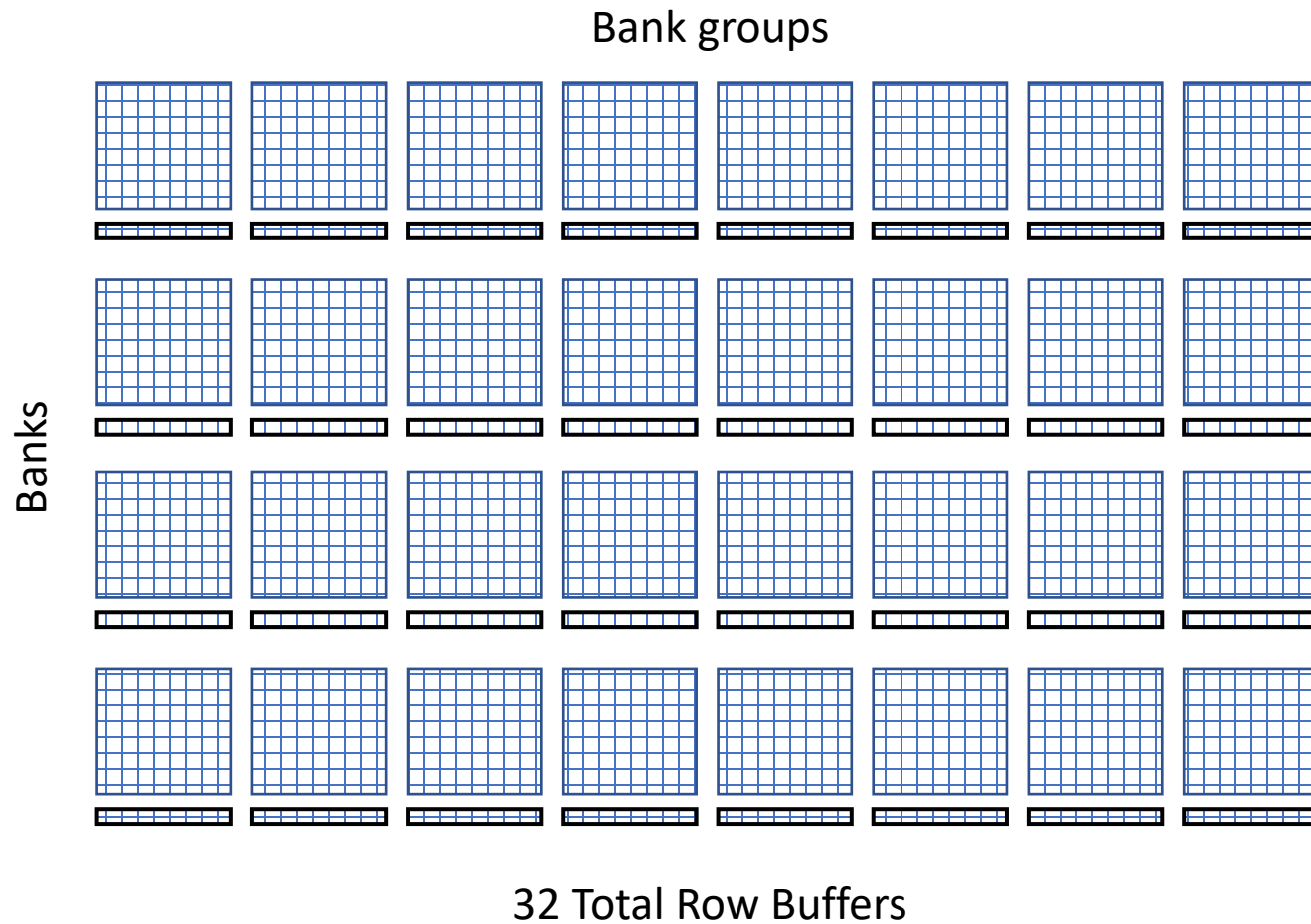
DRAMs maintain an on-chip Refresh Counter to automate the process of restoring bit cell voltage levels





A few fundamental DRAM timings...





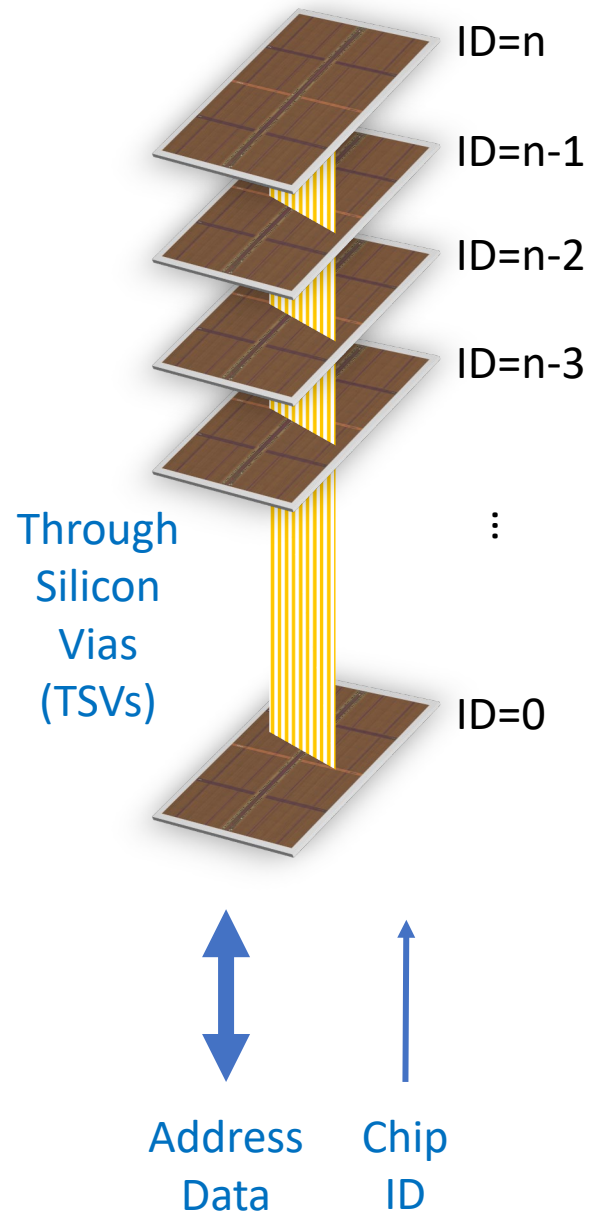
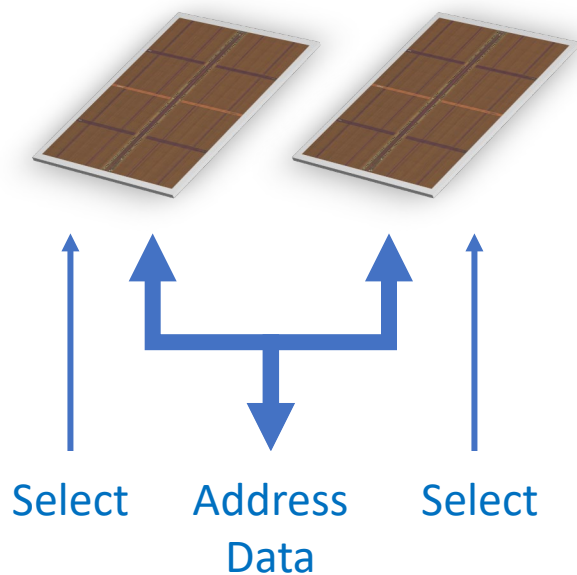
Bank Groups allow for sharing of on-DRAM resources, keeping chip size (and therefore cost) controlled

Now, addressing a particular memory location requires:

Bank Group
Bank
Row
Column



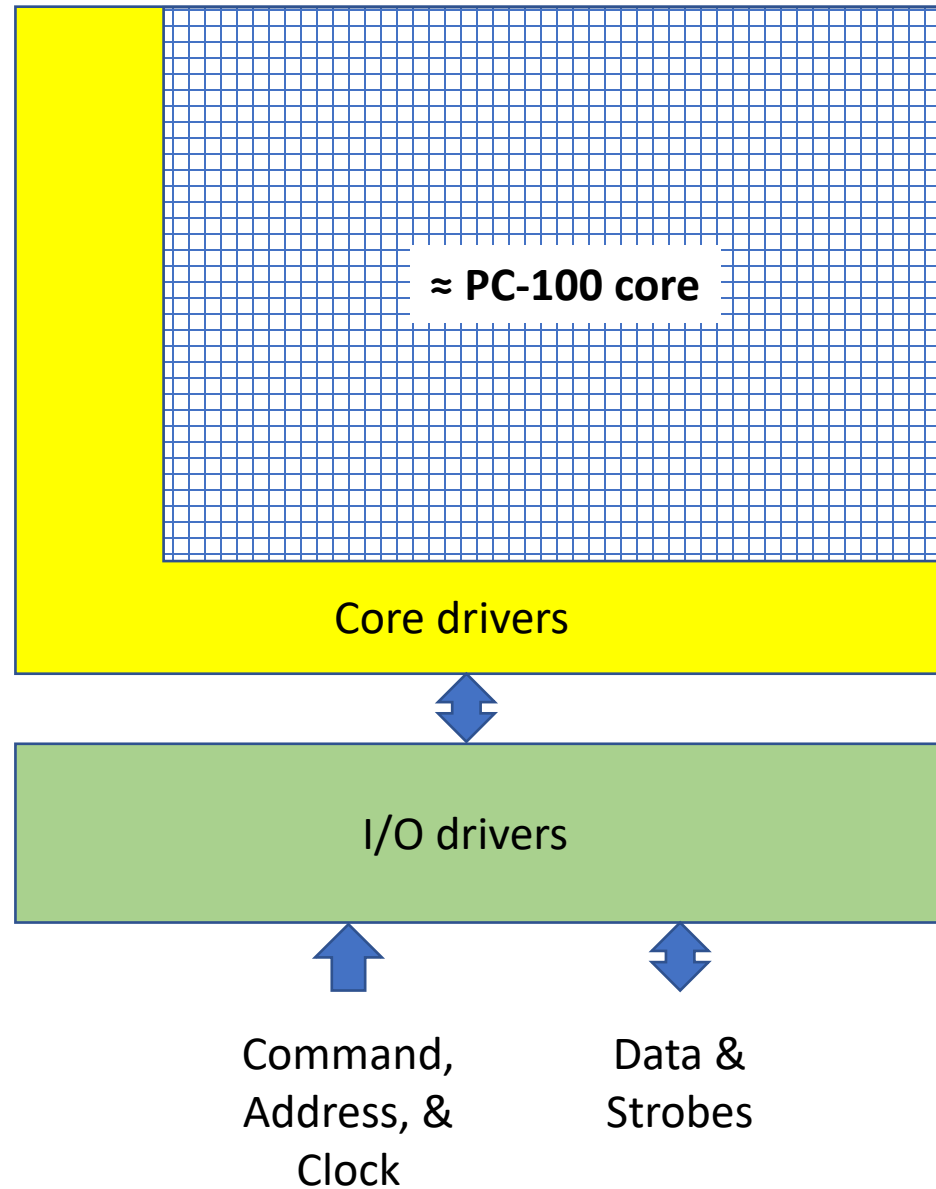
DDP Dual-Die Package



3DS Three Dimensional Stacking

For 3DS, addressing a particular memory location requires:

Chip ID
Bank Group
Bank
Row
Column



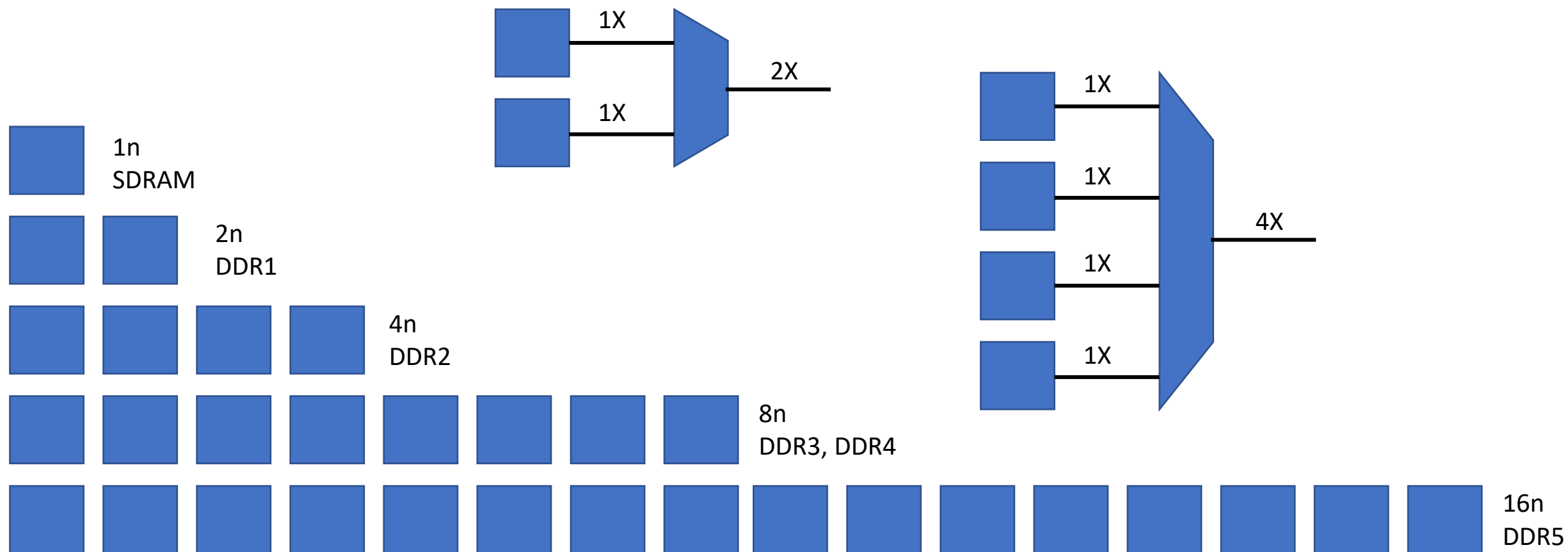
As I said, the core of a DRAM has not changed significantly in a few forevers

The majority of changes have been in the I/O between the host and the DRAM



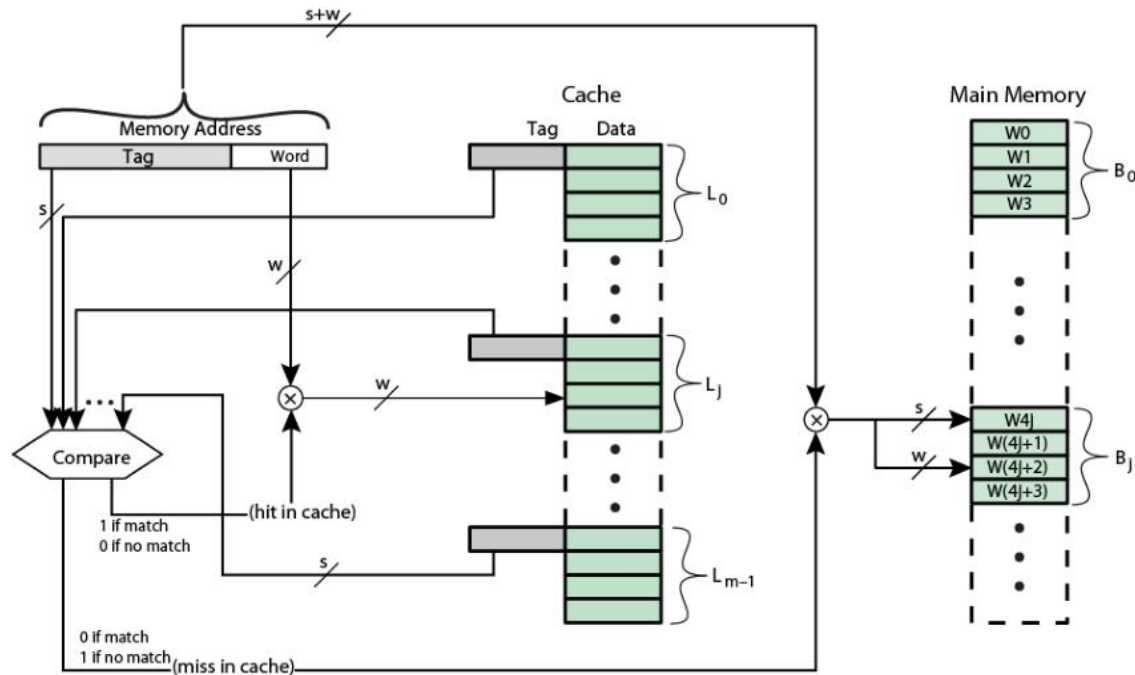
Primary trick #1: PREFETCH

You can keep the core speed the same while doubling the I/O rate by ping-ponging





Cache Line Size Drives the Industry



64-byte cache lines are the standard

This forces memory I/O design to follow rules that provide 64-byte chunks upon request

DRAM I/O widths of x4 and x8 are the most common

A “rank” of memory is the collection of DRAMs accessed simultaneously to provide a wide word to the Host

This drives the math behind “prefetch depth”

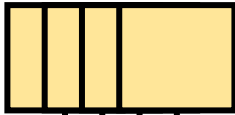
Designs balance DRAM I/O width, per-DRAM capacity to assemble a module



x8

A single x8 device needs **64 transfers** to provide a single cache line

Total solution capacity = 2GB with 16Gb DRAM

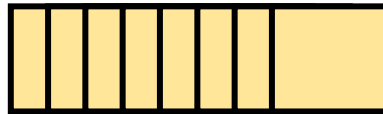


x8

x32

A rank of four x8 devices can provide a cache line in **16 transfers**

Total solution capacity = 8GB with 16Gb DRAMs

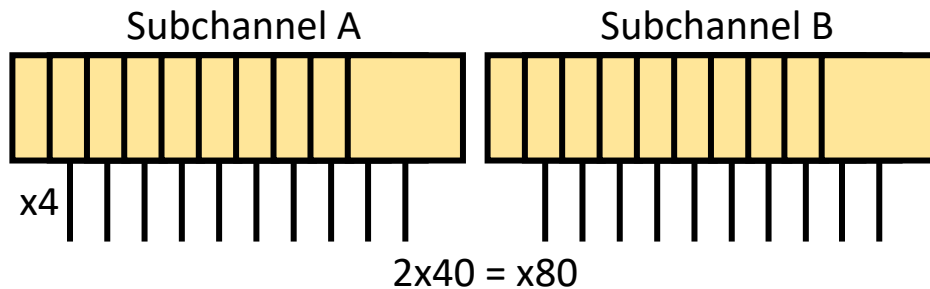


x4

x32

A x4 based solution also supplies a cache lines in **16 transfers**

Total solution capacity = 16GB with 16Gb DRAMs

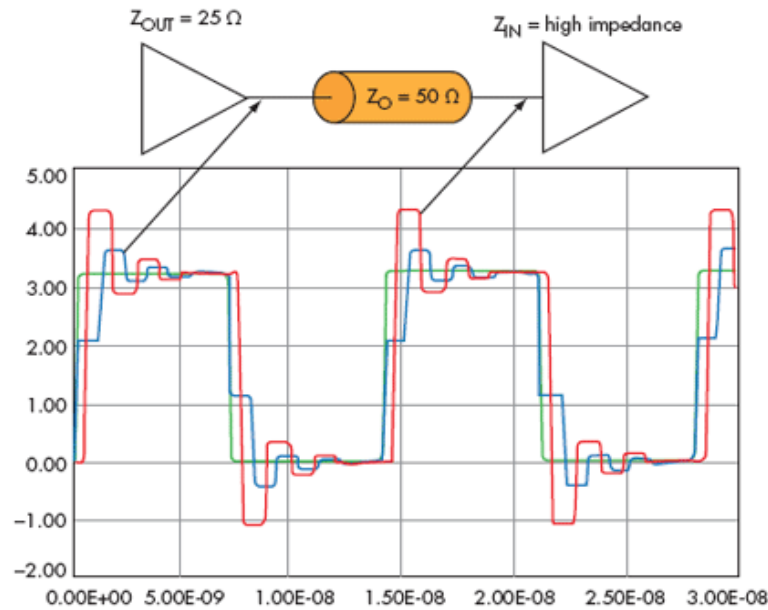


DDR5 server modules have two 40-bit subchannels, each supplies a cache line in **16 transfers**, (32 bit data + 8 ECC)

Capacity = 32GB/rank @ 16Gb (ECC is not counted in capacity)

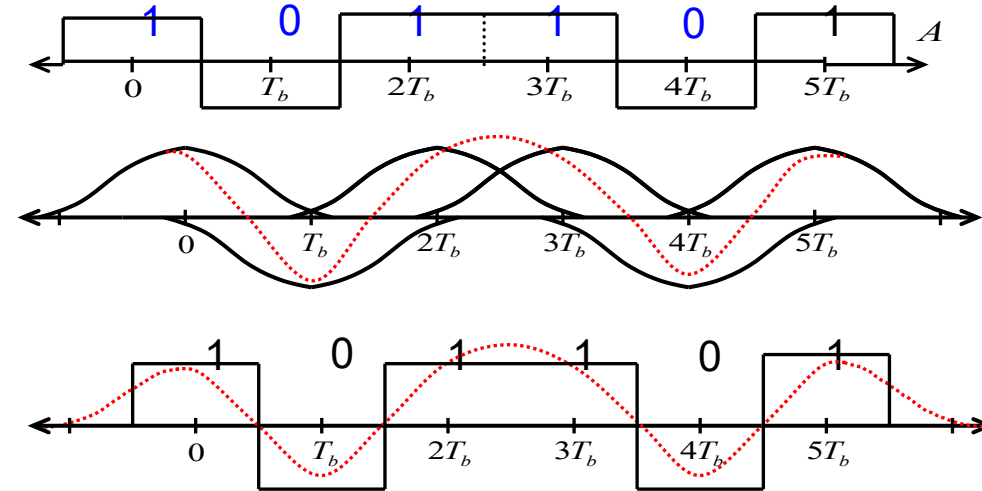


Primary trick #2: Dealing with System I/O Issues

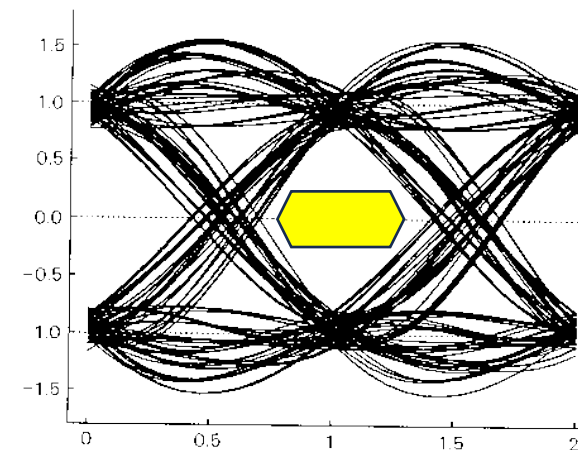


1. In this simulation, reflection is due to an unterminated transmission line. The green signal represents the ideal signal, and the blue and red signals are the driver's side and receiver's input, respectively.

Challenges to reliable data transmission include dealing with reflections from other devices, sockets, and modules

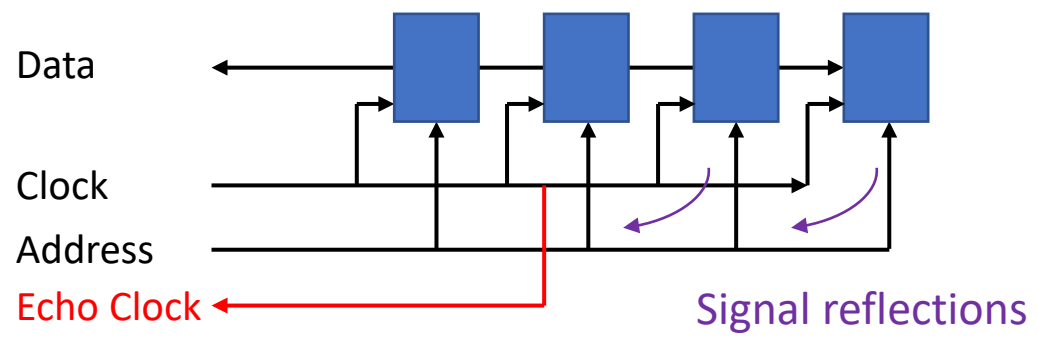
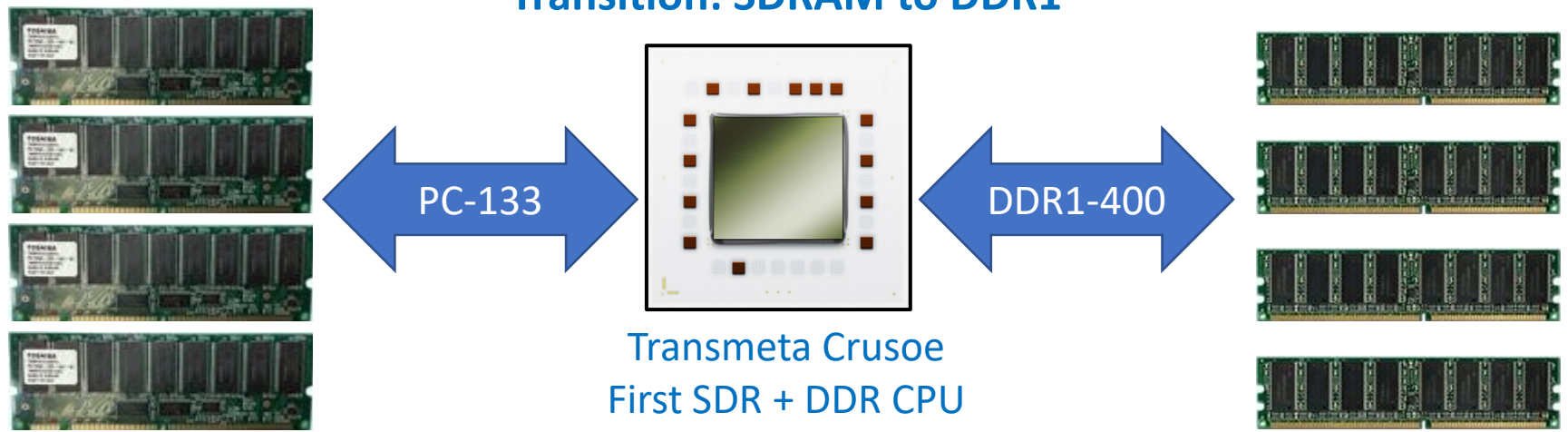


Inter-symbol interference significantly impacts the available data eyes

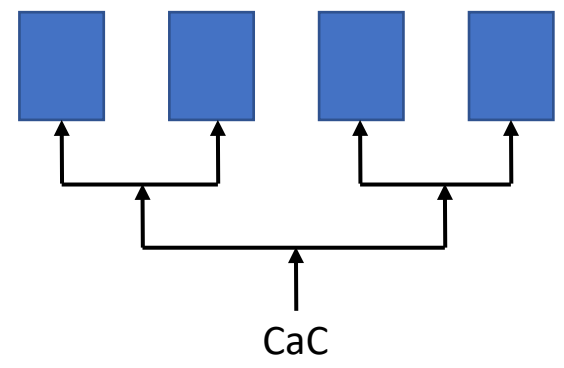
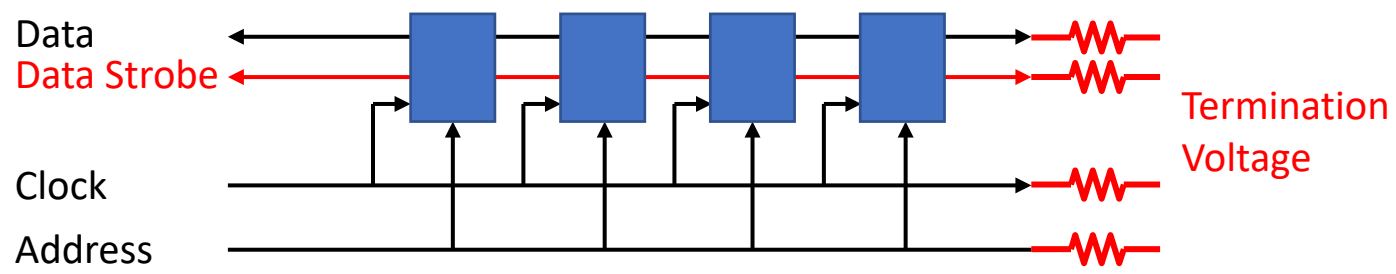




Transition: SDRAM to DDR1



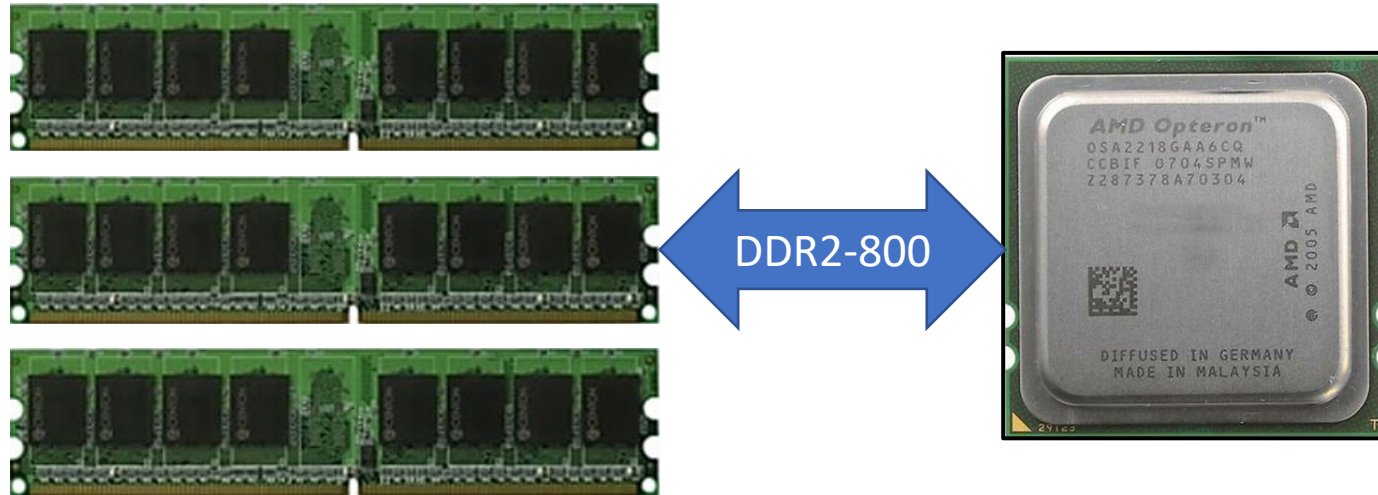
On-DIMM routing of command, address, and clocks used T-branch to match flight times – all DRAMs in same time domain



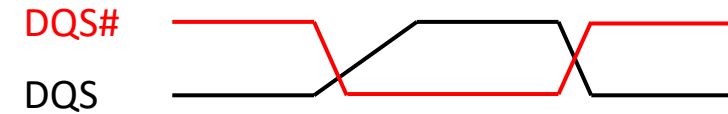
Line-end termination reduced the reflection problem



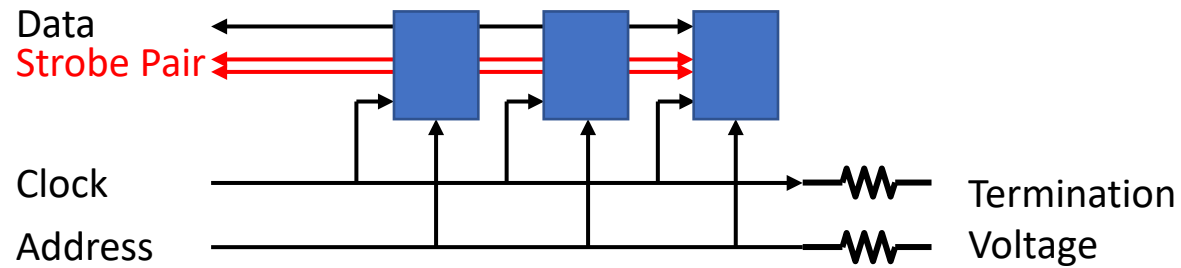
Transition: DDR1 to DDR2



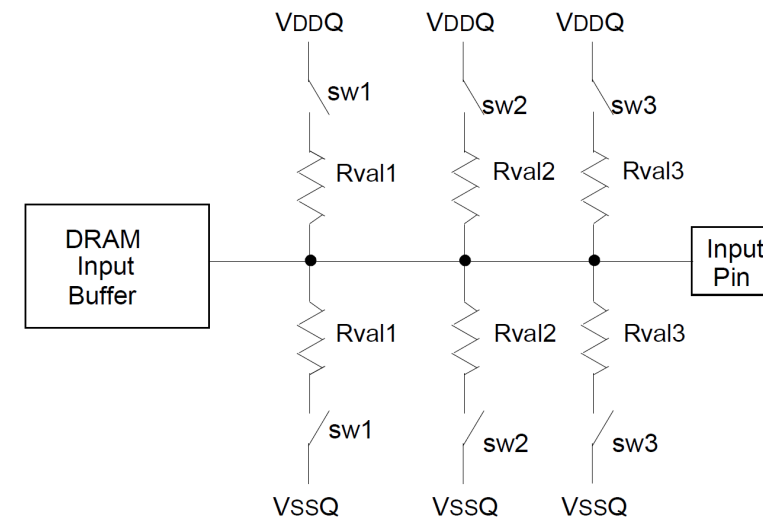
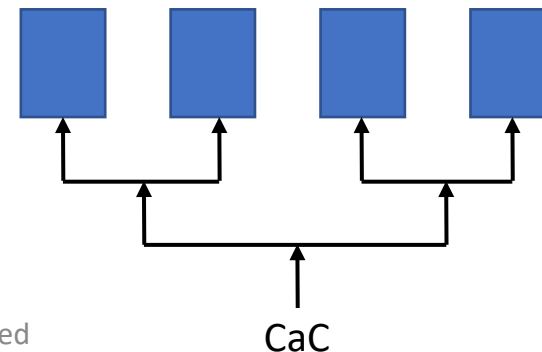
Differential data strobe reduced asymmetry errors



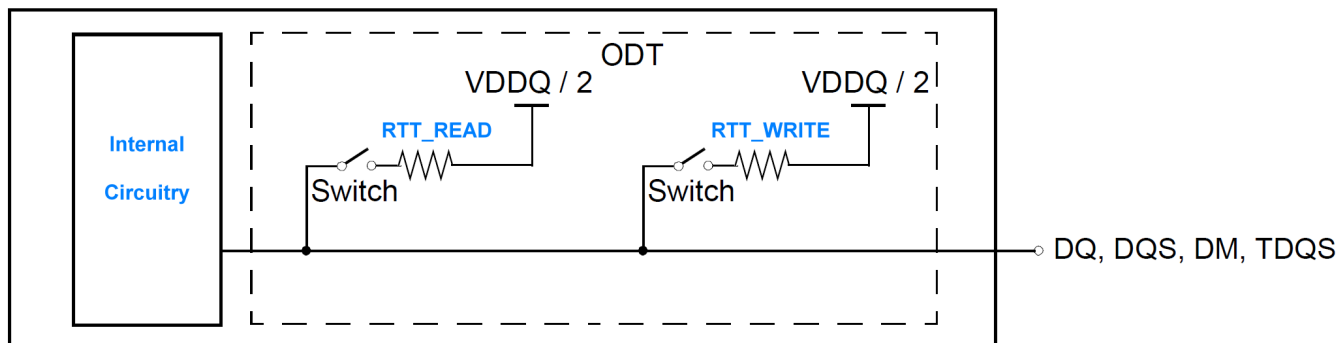
On-die termination reduced transmission line reflections on data and strobes only



DDR2 also used T-branch routing for CaC

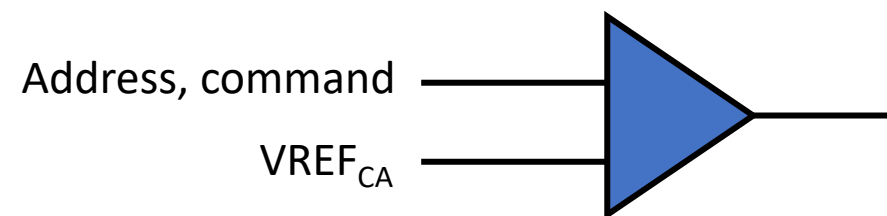
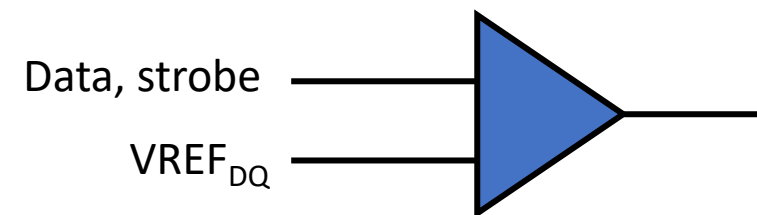


Transition: DDR2 to DDR3



Command-sensitive ODT
improved signal quality

*This figure is reproduced, with permission,
from JEDEC document JESD79-3F, figure 75.*

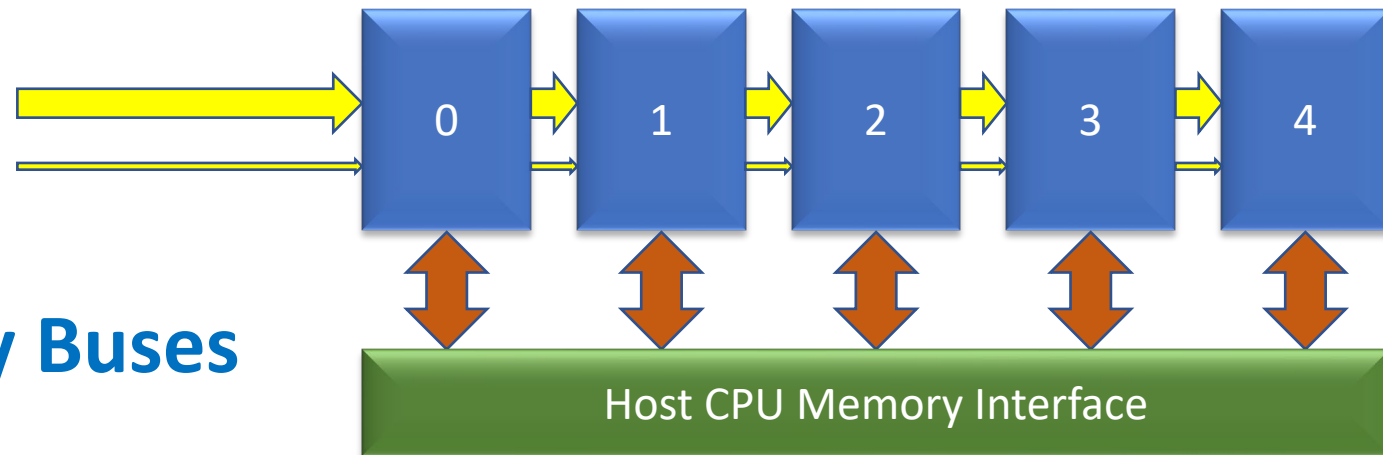


Separate voltage references
for data versus addresses



Fly-by Buses

CMD
Clock

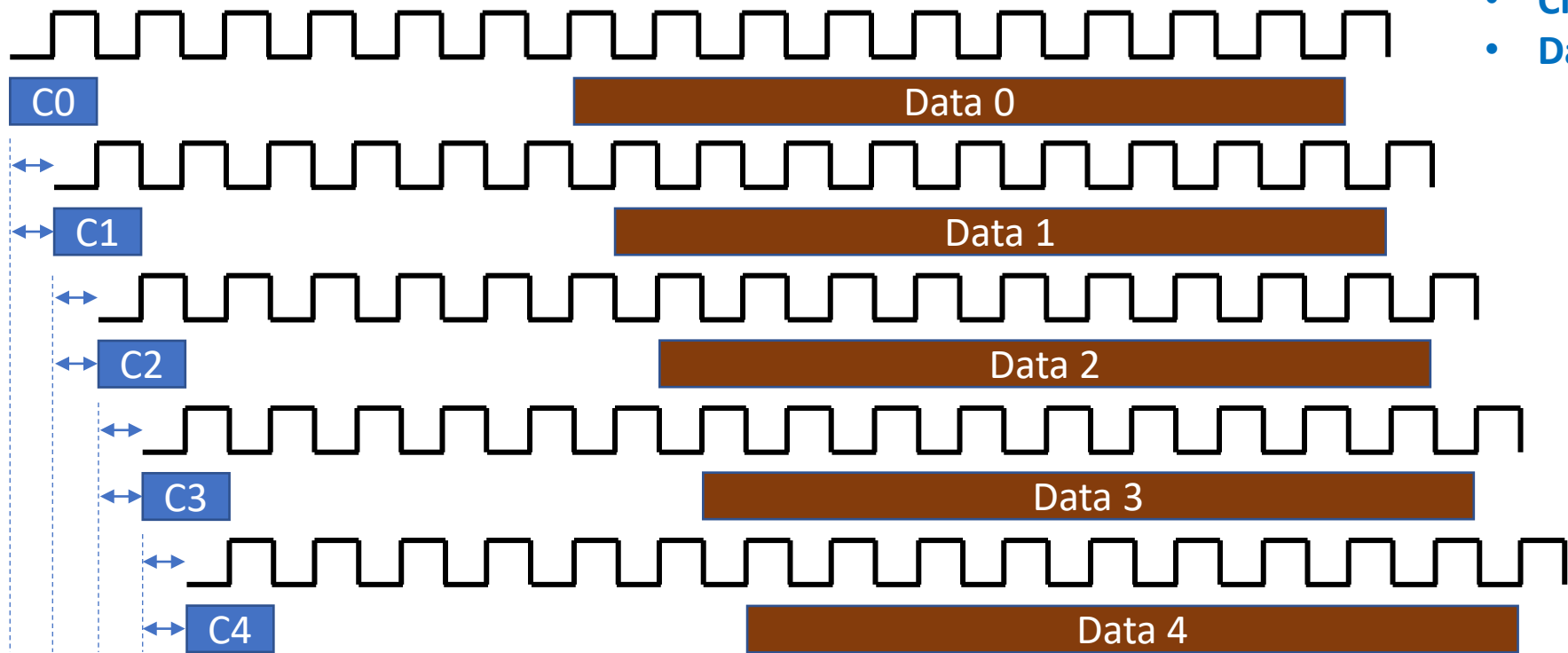


Introduced with DDR3

Memory interface training required to de-skew

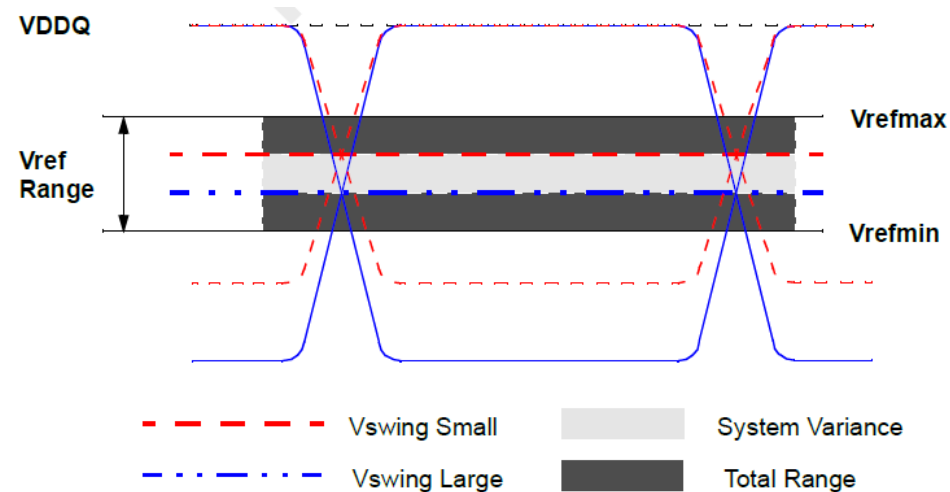
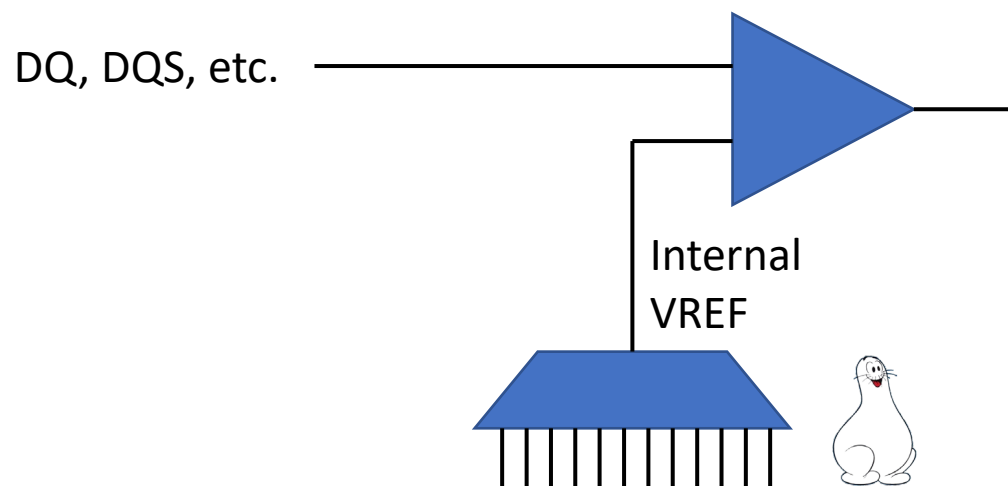
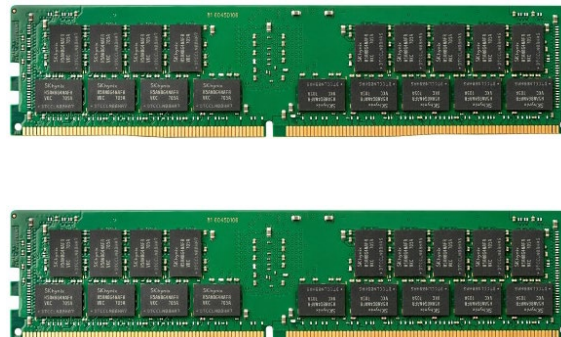
- Clock and address
- Data and strobes

Clock
CMD



Flight time
skews

Transition: DDR3 to DDR4



Shmooing the reference voltage allows tighter calibration

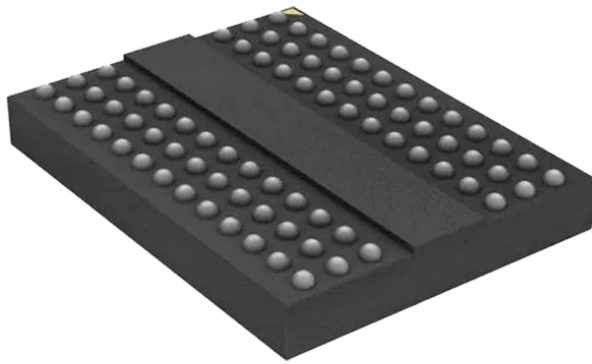
This figure is reproduced, with permission, from JEDEC document JESD79-4C, figure 27.



**JEDEC
STANDARD**

**With that history lesson behind us,
Let's focus on the current mainstream
DRAM generation: DDR5**

DDR5 SDRAM



Designing a mass market solution such as a DRAM requires taking a systems view of the solution

The chip signals need to be routable on a low cost module



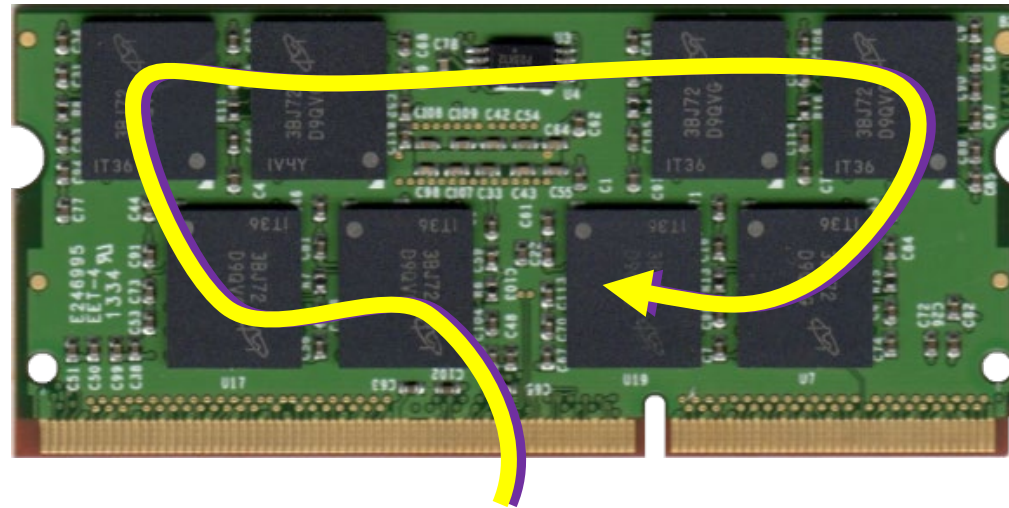
The chip features need to work within a variety of systems

80% of DRAM is sold on memory modules that can plug into sockets

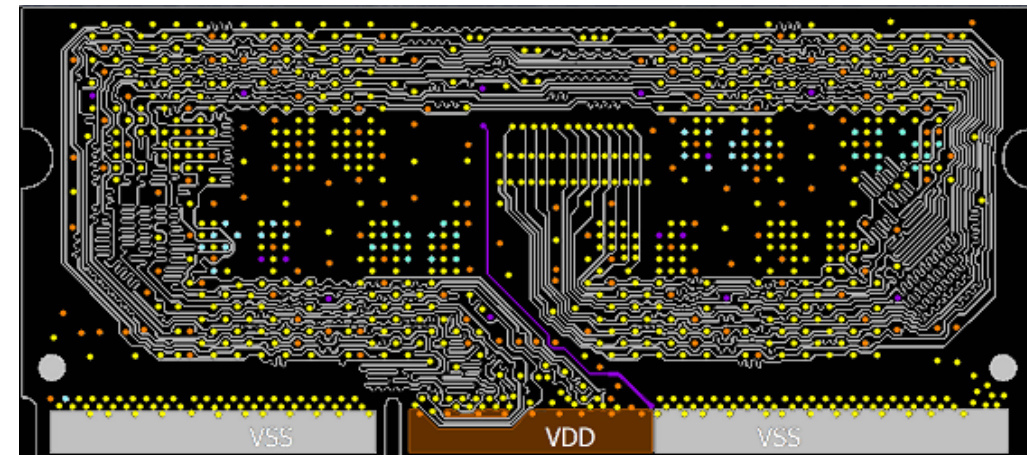
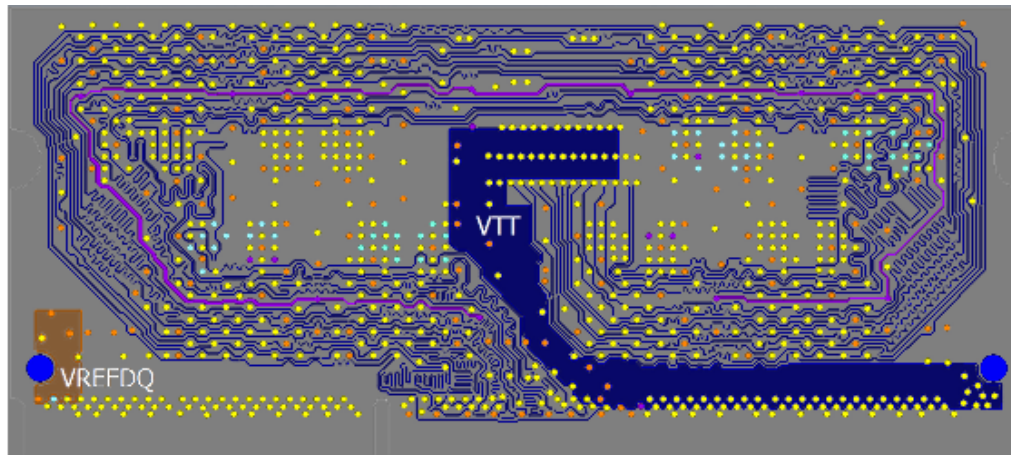
DRAM design is largely driven by the module needs
This training will hop back and forth between DRAM and modules



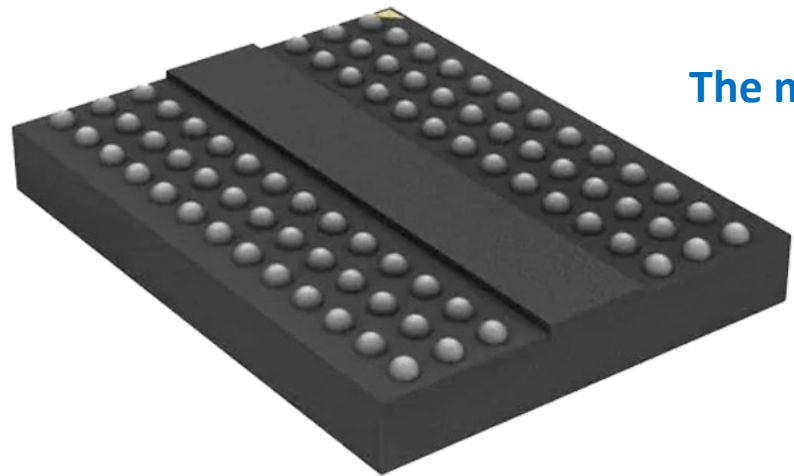
Example: a simple 16-
DRAM module for
notebooks



Address Bus



Just routing the address bus signals requires two layers of printed circuit traces
Increasing the number of signals would increase the number of PCB layers, increasing cost



The module solution requirements drive the design of the DRAM

SDRAMs have a limited number of balls

This enables routing of complex modules

To keep ball count low, signals are **time multiplexed**

This is called the RAS-CAS Protocol for Row and Column functions

AN	1	2	3	4	5	6	7	8	9	10	11	
AL		1	2	3	4	5	6	7	8	9		
A	DNU	LBDQ	VSS	VPP				ZQ	VSS	LBDQS	DNU	A
B		VDD	VDDQ	DQ2				DQ3	VDDQ	VDD		B
C		VSS	DQ0	DQS_t				DM_n, TDQS_t	DQ1	VSS		C
D		VDDQ	VSS	DQS_c				TDQS_c	VSS	VDDQ		D
E		VDD	DQ4	DQ6				DQ7	DQ5	VDD		E
F		VSS	VDDQ	VSS				VSS	VDDQ	VSS		F
G		CA_ODT	MIR	VDD				CK_t	VDDQ	TEN		G
H		ALERT_n	VSS	CS_n				CK_c	VSS	VDD		H
J		VDDQ	CA4	CA0				CA1	CA5	VDDQ		J
K		VDD	CA6	CA2				CA3	CA7	VDD		K
L		VDDQ	VSS	CA8				CA9	VSS	VDDQ		L
M		CAI	CA10	CA12				CA13	CA11	RESET_n		M
N	DNU	VDD	VSS	VDD				VPP	VSS	VDD	DNU	N

This figure is reproduced, with permission, from JEDEC document JESD79-5B, table 1.

DDR5 Command Truth Table



Flash Memory Summit

Function	Abbrevia- tion	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
		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
		H	V	C3	C4	C5	C6	C7	C8	C9	C10	V	H	WR Partial=L	V	CID3
Read	RD	L	H	L	H	H	H	BL*=L	BA0	BA1	BG0	BG1	BG2	CID0	CID1	CID2
		H	C2	C3	C4	C5	C6	C7	C8	C9	C10	V	H	V	V	CID3
Precharge	PREpb	L	H	H	L	H	H	CID3	BA0	BA1	BG0	BG1	BG2	CID0	CID1	CID2

*This figure is reproduced, with permission,
from JEDEC document JESD79-5B, table 31.*

“RAS”

“CAS”

“CAS”

ACTIVATE

Core read

READ

Data → I/O

WRITE

Data ← I/O

PRECHARGE

Core restored

Chip ID +
Bank Group +
Bank Address +
Row

Chip ID +
Bank Group +
Bank Address +
Column

Chip ID +
Bank Group +
Bank Address +
Column

Chip ID +
Bank Group +
Bank Address

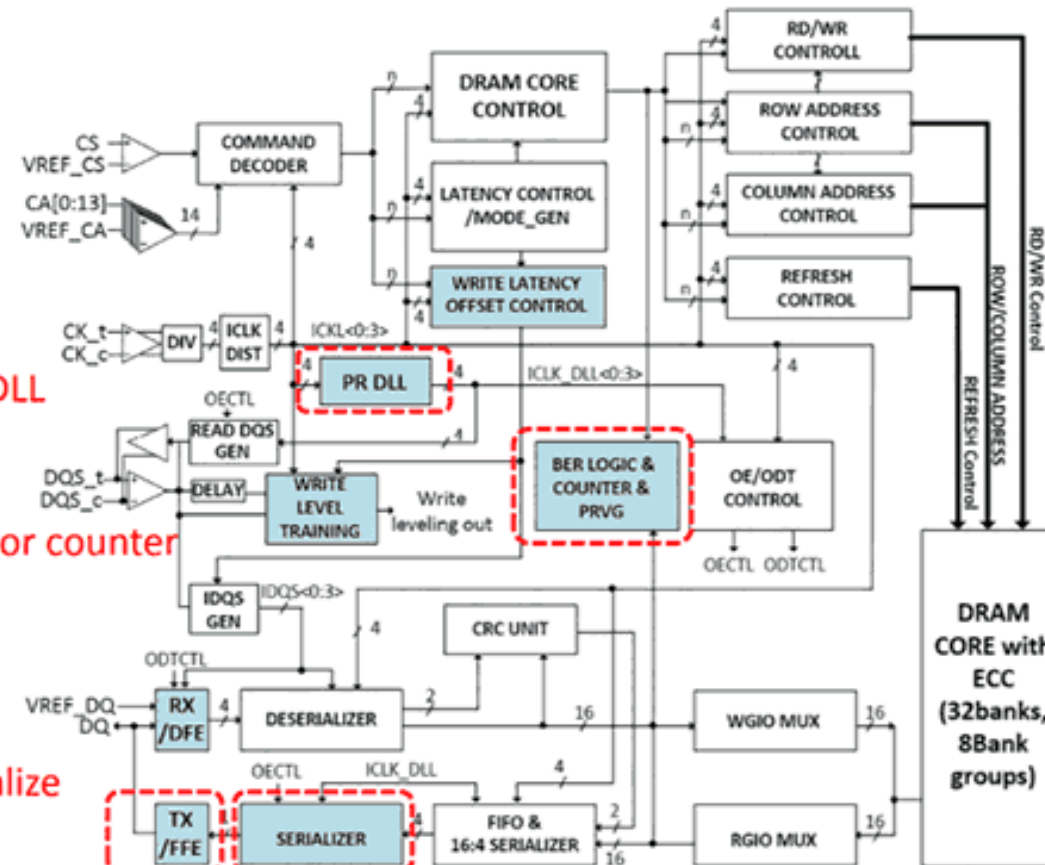


A nice block diagram from SK Hynix

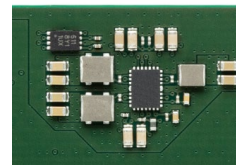
DDR5 Block diagram

This gives a sense for the building blocks we will explore

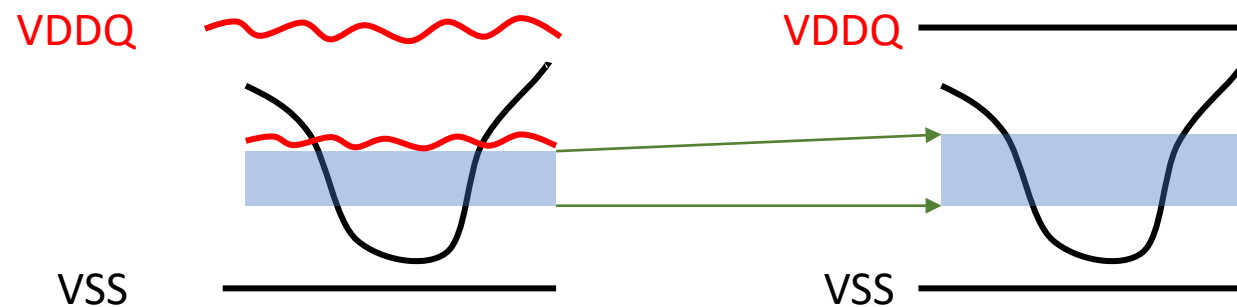
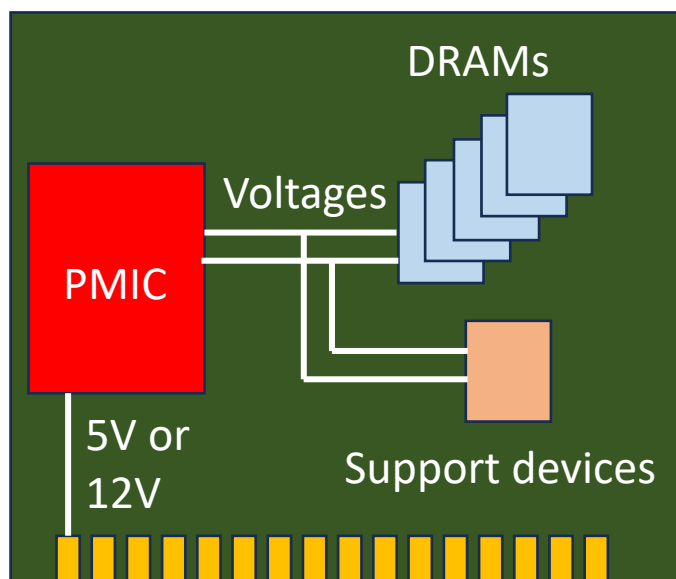
- Phase rotator DLL
- BER Logic & Error counter
- FFE for Tx equalize
- 4:1 serializer



Transition: DDR4 to DDR5



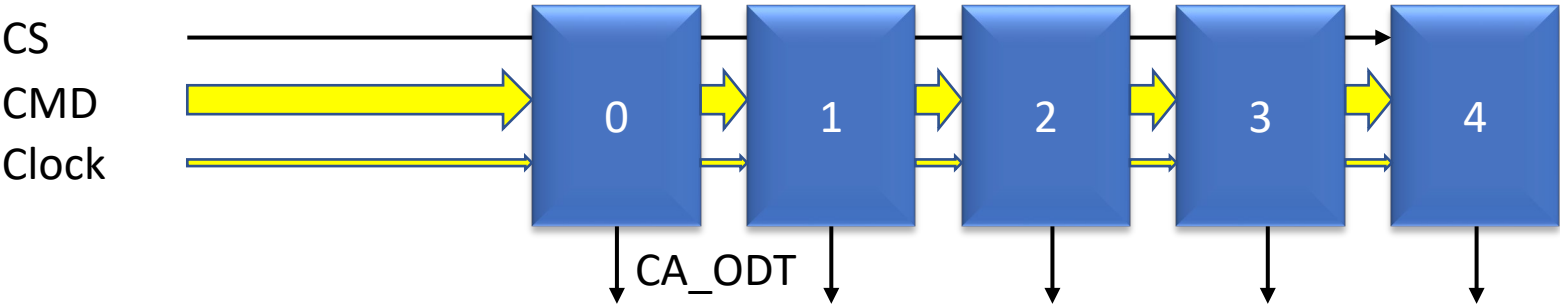
Back side: on-DIMM voltage regulation



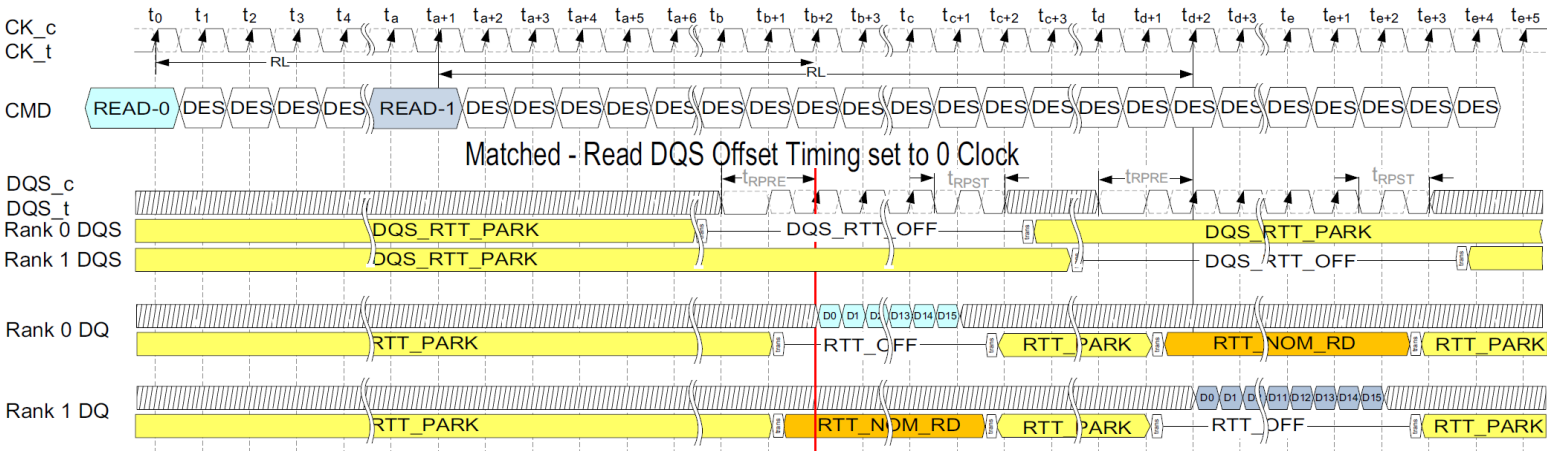
Noise on the voltage rails results in reduction of input sensitivity

Noise reduction improves data integrity

Command, address, clock, chip select ODT



Clean command, address, clock, and chip select signals are managed via mode register settings
CA_ODT pin selects a “group” of termination settings for command/address



This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 32.

Mode Registers

Clean data and strobe signals are managed by command-driven ODT
Termination values set via mode registers

32	RFU	CA_ODT Strap Value	CS ODT	CK ODT
33	RFU		DQS_RTT_PARK	CA ODT
34	RFU		RTT_WR	RTT_PARK
35	RFU		RTT_NOM_RD	RTT_NOM_WR

Managing DRAM Optional Features & Settings



Flash Memory Summit

MR#	OP[7]	OP[6]	OP[5]	OP[4]	OP[3]	OP[2]	OP[1]	OP[0]
0	RFU	CAS Latency (RL)				Burst Length		
1	PDA Select ID				PDA Enumerate ID			
2	Internal Write Timing	Reserved	Device 15 MPMS	CS Assertion Duration (MPC)	Max Power Saving Mode (MPMS)	2N Mode	Write Leveling Training	Read Preamble Training
3	Write Leveling Internal Cycle Alignment - Upper Byte				Write Leveling Internal Cycle Alignment - Lower Byte			
4	TUF	RFU	Wide Range (Optional)	Refresh tRFC Mode	Refresh Interval Rate Indicator	Minimum Refresh Rate		
5	Pull-Down Output Driver Impedance		DM Enable	TDQS Enable	PODTM Support	Pull-up Output Driver Impedance		Data Output Disable
6	tRTP				Write Recovery Time			
7	RFU					(Optional) Write Leveling Internal +0.5tCK Alignment Offset - Upper Byte		(Optional) Write Leveling Internal +0.5tCK Alignment Offset - Lower Byte

DDR5 Mode Registers

This figure is reproduced, with permission, from JEDEC document JESD79-5B, table 24.

Byte Number		DDR5 Function Described	Notes
0	0x000	Number of Bytes in SPD Device and Beta Level	
1	0x001	SPD Revision for Base Configuration Parameters	
2	0x002	Key Byte / Host Bus Command Protocol Type	
3	0x003	Key Byte / Module Type	
4	0x004	First SDRAM Density and Package	1
5	0x005	First SDRAM Addressing	1
6	0x006	First SDRAM I/O Width	1
7	0x007	First SDRAM Bank Groups & Banks Per Bank Group	1
8	0x008	Second SDRAM Density and Package	1
9	0x009	Second SDRAM Addressing	1
10	0x00A	Second SDRAM I/O Width	1
11	0x00B	Second SDRAM Bank Groups & Banks Per Bank Group	1
12	0x00C	SDRAM BL32 & Post Package Repair	1
13	0x00D	SDRAM Duty Cycle Adjuster & Partial Array Self Refresh	1
14	0x00E	SDRAM Fault Handling	1
15	0x00F	Reserved	
16	0x010	SDRAM Nominal Voltage, VDD	1
17	0x011	SDRAM Nominal Voltage, VDDQ	1
18	0x012	SDRAM Nominal Voltage, VPP	1
19	0x013	SDRAM Timing	1
20	0x014	SDRAM Minimum Cycle Time (t _{CKAVGmin}), Least Significant Byte	1
21	0x015	SDRAM Minimum Cycle Time (t _{CKAVGmin}), Most Significant Byte	1
22	0x016	SDRAM Maximum Cycle Time (t _{CKAVGmax}), Least Significant Byte	1

This figure is reproduced, with permission, from JEDEC document JESD400-5, table 1.

DRAM mode registers are a mix of mandatory mode settings, optional features, and DRAM feedback bits

During initialization, DRAM power may not be on so MRs are not available

The SPD chip contains information about the DRAMs on the module, and helps track options as DDR5 evolves

DDR5 SPD

(DDR5): Byte 14 (0x00E): SDRAM Fault Handling and Temperature Sense

This byte defines support for SDRAM fault handling features and for wide on-die temperature sensing range (see MR4). This value comes from the DDR5 SDRAM data sheet, JESD79-5.

SDRAM Fault Handling Features and Temperature Sense			
Bits 7~4			
Reserved; must be coded as 0000			
Bit 3	Bit 2	Bit 1	Bit 0
Wide Temperature Sense	x4 RMW/ECS Writeback Suppression	x4 RMW/ECS Writeback Suppression MR Selector	Bounded Fault
0: Wide temperature sense and reporting not supported 1: Wide temperature sense and reporting supported	0: Writeback suppression not supported 1: Writeback suppression supported	0: Writeback suppression control in MR9 1: Writeback suppression control in MR15	0: Bounded Fault not supported 1: Bounded Fault supported
Notes: ECS = Error Check Scrub RMW = Read-Modify-Write MR = Mode Register			

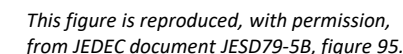
This figure is reproduced, with permission, from JEDEC document JESD400-5, section 8.1.15.

Initialization and Training Modes

*This figure is reproduced, with permission,
from JEDEC document JESD79-5B, table 81.*

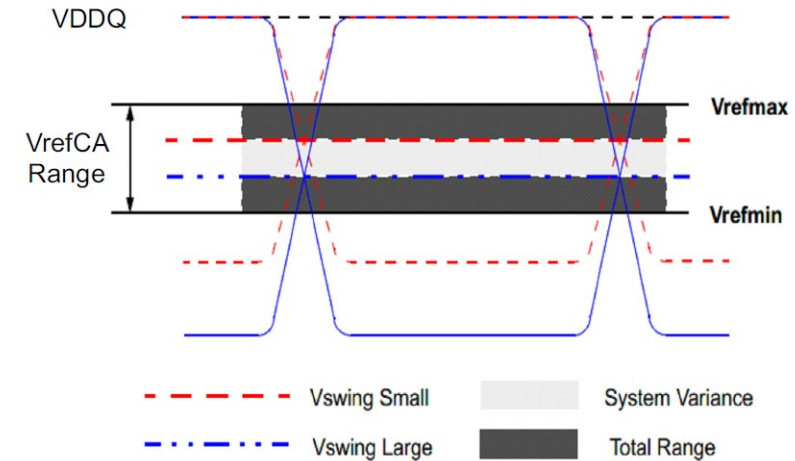
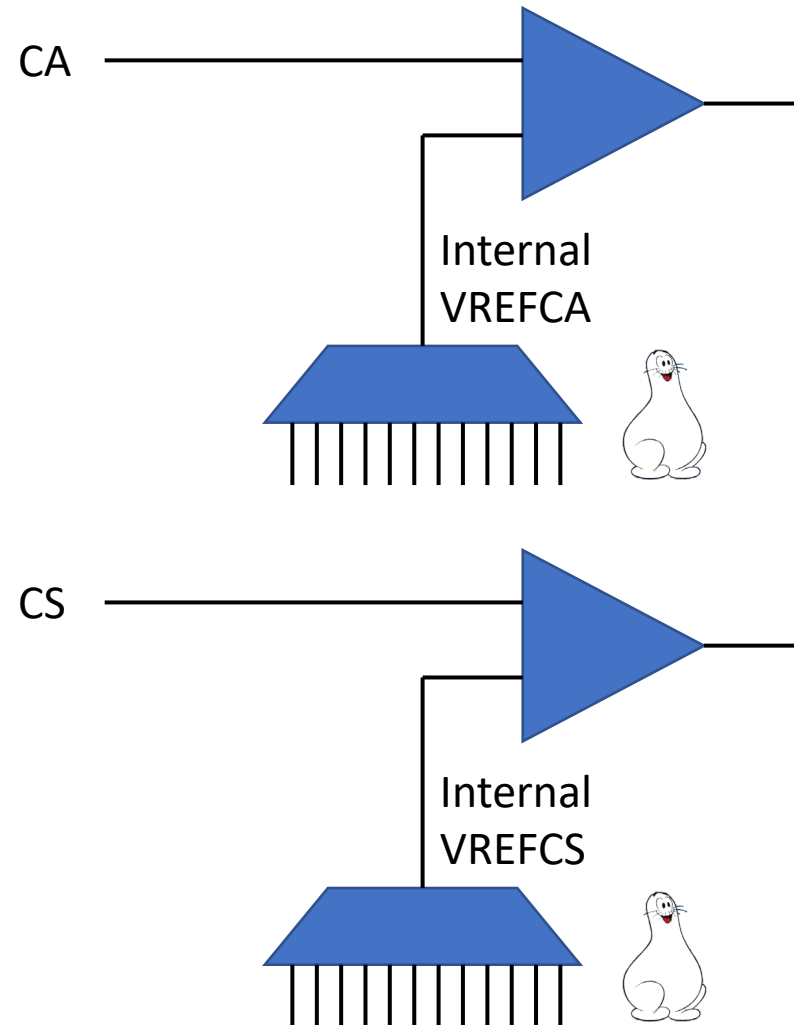
A diagram illustrating data flow. On the left, the text "CaC" is followed by a large yellow arrow pointing to a blue square box labeled "DRAM". A black arrow points downwards from the bottom of the "DRAM" box.

DQ = four CK sampling (CS toggle)
DQ = XOR(CA[13:0])



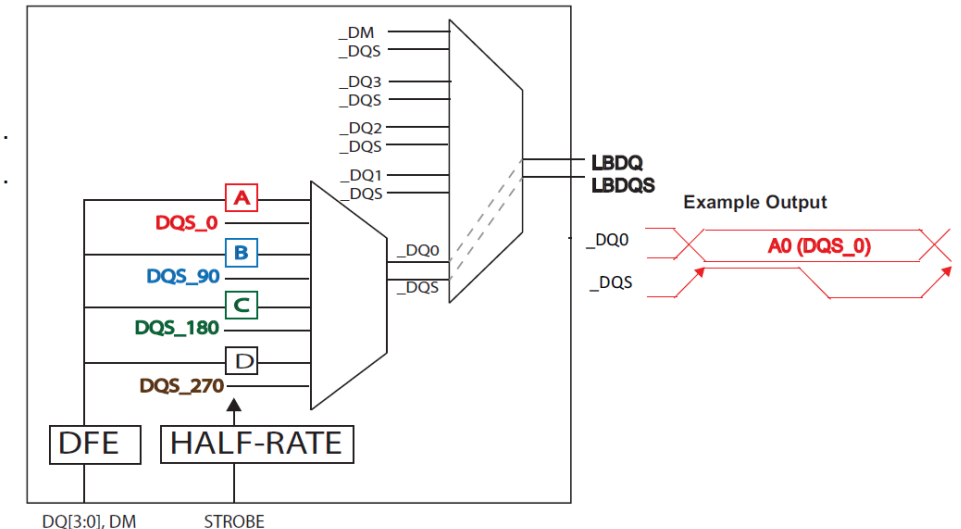
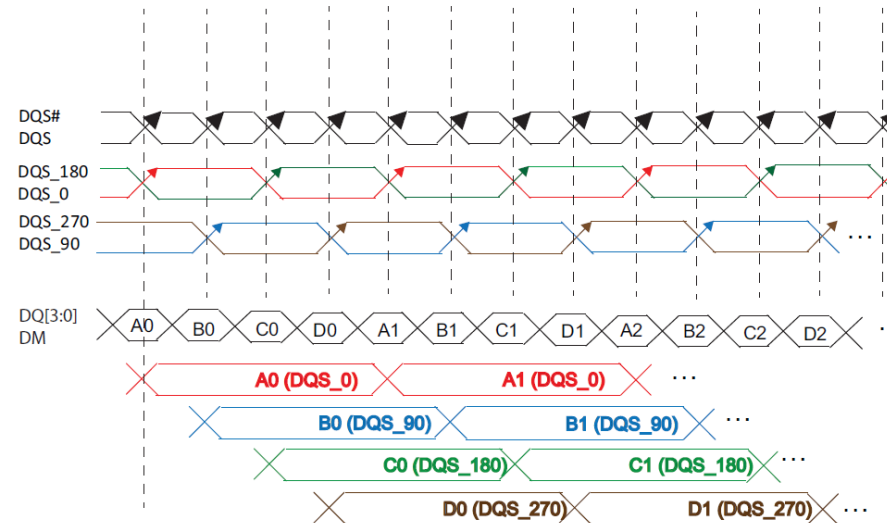
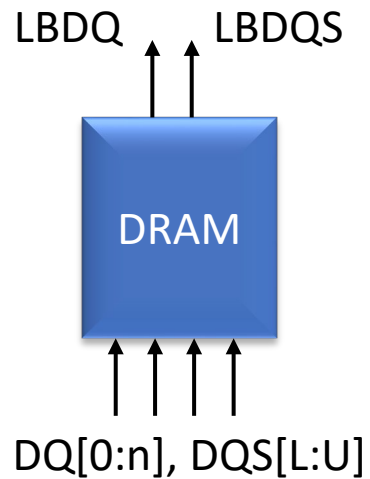


Chip Select Training Mode (CSTM) and Command/Address Training Mode (CATM) are similar to DQ VREF training in DDR4



*This figure is reproduced, with permission,
from JEDEC document JESD79-5B, figure 114.*

DDR5 Loopback outputs may be used during initialization, or even at runtime, to check the integrity of the data bus interface – fed to external logic such as Registering Clock Driver (RCD) to check



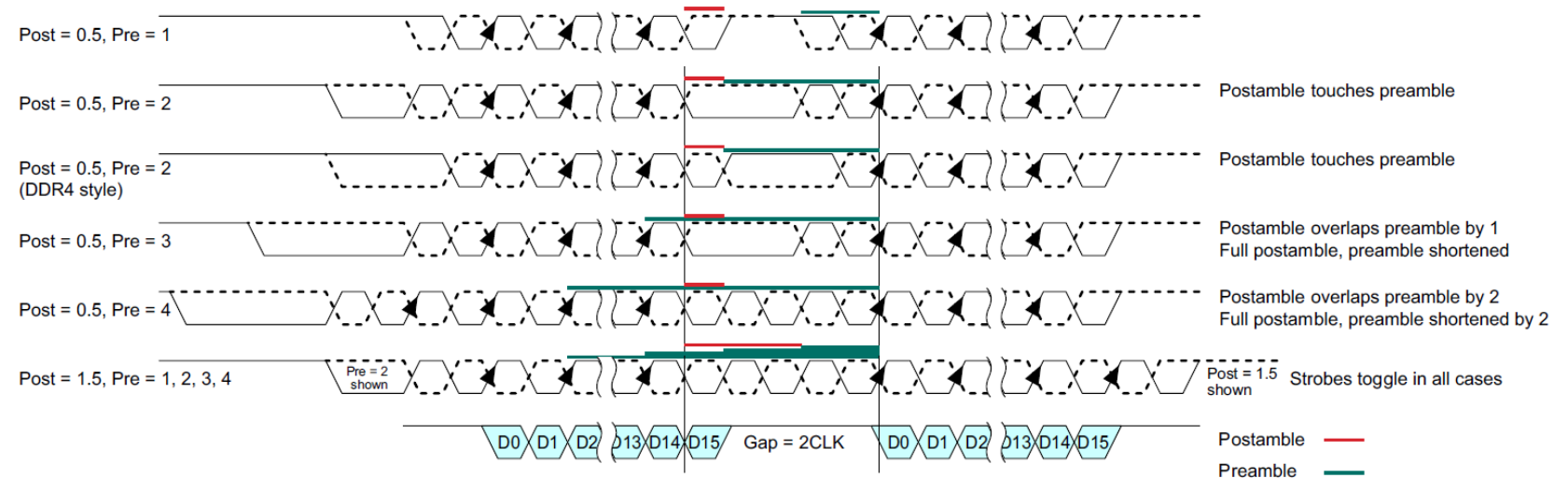
This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 163.



Function	Register Type	Operand	Data
Read Preamble Settings	R/W	OP[2:0]	000B: 1 tCK - 10 Pattern 001B: 2 tCK - 0010 Pattern 010B: 2 tCK - 1110 Pattern (DDR4 Style) 011B: 3 tCK - 000010 Pattern 100B: 4 tCK - 00001010 Pattern 101B: Reserved 110B: Reserved 111B: Reserved
Write Preamble Settings	R/W	OP[4:3]	00B: Reserved 01B: 2 tCK - 0010 Pattern (Default) 10B: 3 tCK - 000010 Pattern 11B: 4 tCK - 00001010 Pattern
RFU	RFU	OP[5]	RFU
Read Postamble Settings	R/W	OP[6]	0B: 0.5 tCK - 0 Pattern 1B: 1.5 tCK - 010 Pattern
Write Postamble Settings	R/W	OP[7]	0B: 0.5 tCK - 0 Pattern 1B: 1.5 tCK - 000 Pattern

As frequencies get higher, or system routing more complex, additional edges are needed on data strobes to precondition the lines

This figure is reproduced, with permission, from JEDEC document JESD79-5B, section 3.5.10.

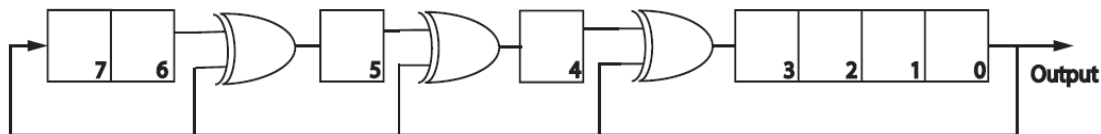


The tradeoff is potentially bigger bubbles on the data bus when streaming back to back operations

This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 20.

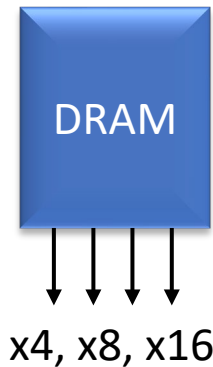
4.17.2 LFSR Pattern Generation

The LFSR is an 8-bit Galois LFSR. The polynomial for the Galois LFSR is $x^8+x^6+x^5+x^4+1$. The figure below shows the logic to implement the LFSR. The numbered locations within the shift register show the mapping of the seed/state positions within the register. There are two instances of the same LFSR polynomial. These two instances will have unique seeds/states and supply patterns to any of the DQ outputs.



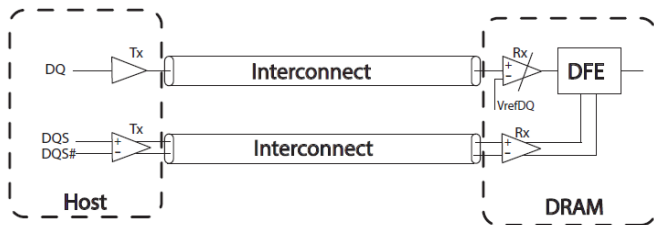
This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 90.

Various patterns may be selected
This is a typical pattern set



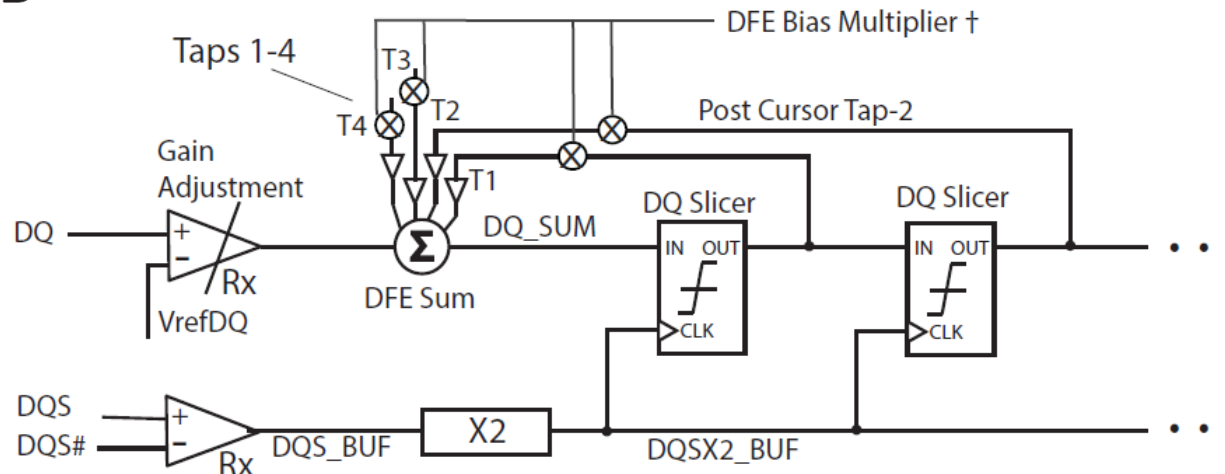
DDR5 read calibration is enhanced by generating a continuous burst pattern through a linear feedback shift register (LFSR)

Pin	Invert	LFSR	Bit Sequence															
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DQL0	0 (No)	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	0
DQL1	0 (No)	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0
DQL2	0 (No)	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0
DQL3	0 (No)	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0
DQL4	0 (No)	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0
DQL5	0 (No)	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0
DQL6	0 (No)	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0
DQL7	0 (No)	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0
DQU0	1 (Yes)	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	0	1
DQU1	1 (Yes)	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1
DQU2	1 (Yes)	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1
DQU3	1 (Yes)	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1
DQU4	1 (Yes)	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1
DQU5	1 (Yes)	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1
DQU6	1 (Yes)	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1
DQU7	1 (Yes)	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1



This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 141.

DFE optionally has up to 4 taps on each signal



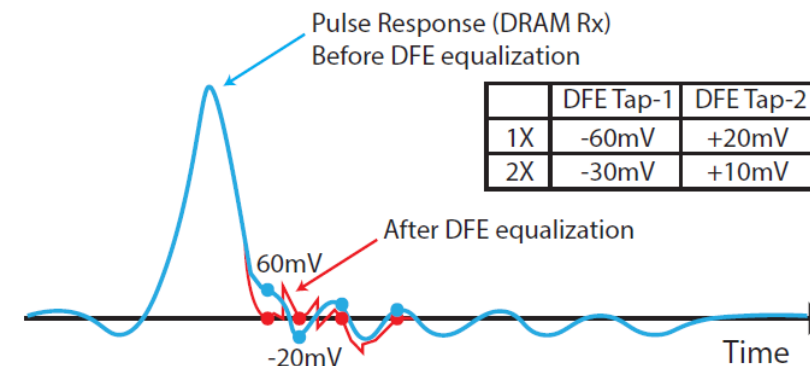
This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 143.

128	RFU			DQLO DFE Gain Bias - See MR for encoding details		
129	DQLO DFE Tap-1 Bias - See MR for encoding details					
130	DQLO DFE Tap-2 Bias - See MR for encoding details					
131	DQLO DFE Tap-3 Bias - See MR for encoding details					
132	DQLO DFE Tap-4 Bias - See MR for encoding details					
133	DQLO IBCLK Sign	RFU	DQLO DCA for IBCLK	DQLO QCLK Sign	RFU	DQLO DCA for QCLK
134	DQLO VREFDQ Sign	DQLO VREFDQ Offset		DQLO QBCLK Sign	RFU	DQLO DCA for QBCLK
135	RFU					
136	RFU			DQL1 DFE Gain Bias - See MR for encoding details		
137	DQL1 DFE Tap-1 Bias - See MR for encoding details					
138	DQL1 DFE Tap-2 Bias - See MR for encoding details					
139	DQL1 DFE Tap-3 Bias - See MR for encoding details					
140	DQL1 DFE Tap-4 Bias - See MR for encoding details					
141	DQL1 IBCLK Sign	RFU	DQL1 DCA for IBCLK	DQL1 QCLK Sign	RFU	DQL1 DCA for QCLK
142	DQL1 VREFDQ Sign	DQL1 VREFDQ Offset		DQL1 QBCLK Sign	RFU	DQL1 DCA for QBCLK
143	RFU					

This figure is reproduced, with permission, from JEDEC document JESD79-5B, table 81.

Mode Registers control the DFE settings on a per bit basis

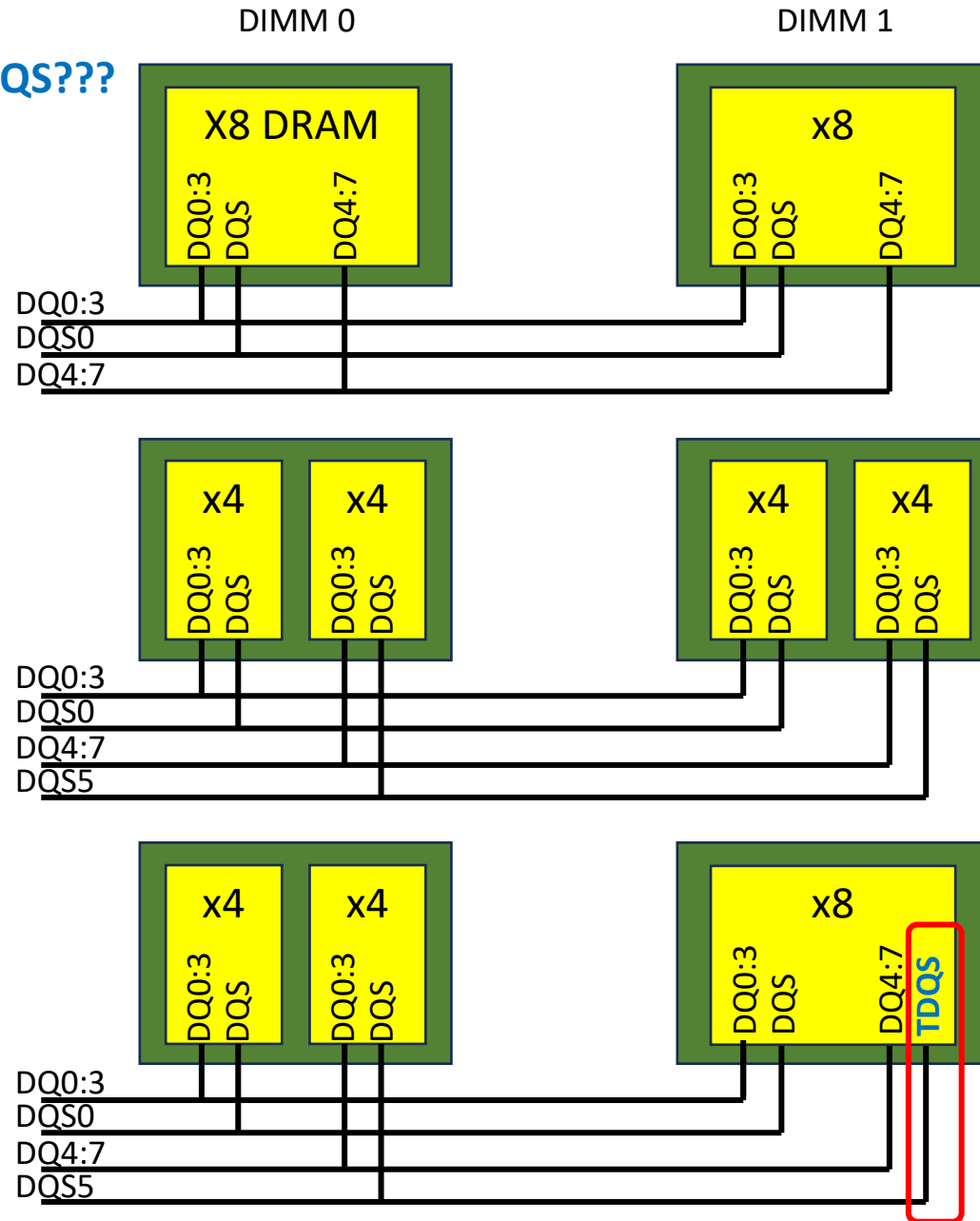
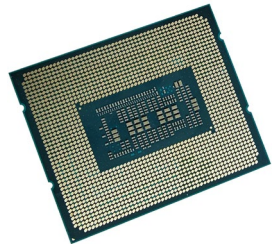
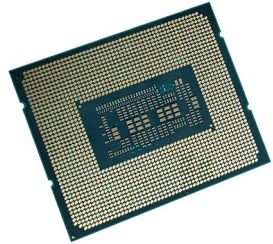
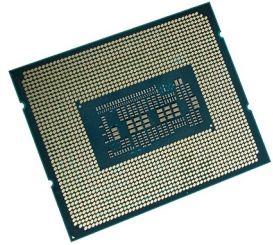
The impact of DFE is significantly improved data eye tracking



This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 142.



What the Heck is TDQS???



Two x8-based modules? No problem
All 8 data bits are associated with a single DQS pair – ODT works as normal

Two x4-based modules? No problem
Each 4 data bits are associated with a DQS pair – ODT works as normal

Mixing x4-based and x8-based modules on the same bus? BIG PROBLEM
One DQS pair gets terminated at the x4 DRAMs but would be left floating at the x8

TDQS on x8 DRAMs allows termination of the “floating” strobe pair

Filling a region of memory with a fill pattern is common

“Fill with 0” was considered BUT while this might work for bulk data, it likely would not work for ECC or metadata

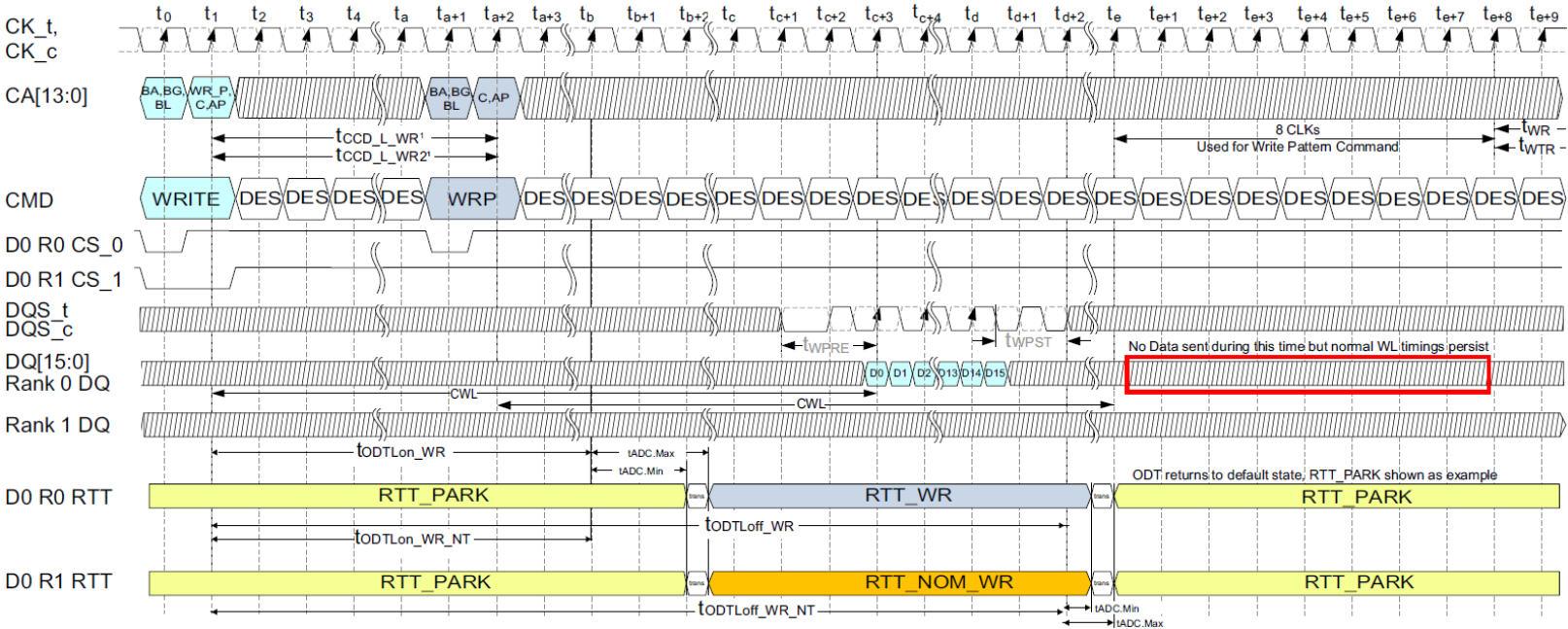
Write Pattern command allows a burst of a set value loaded in mode registers into a column

SDRAM CONFIG	BL16 x16	BL16 x8	BL16 x4	BL32 x4
UI	0-15	0-15	0-15	0-31
DQL0 / DQ0	OP0	OP0	OP0	OP0
DQL1 / DQ1	OP1	OP1	OP1	OP1
DQL2 / DQ2	OP2	OP2	OP2	OP2
DQL3 / DQ3	OP3	OP3	OP3	OP3
DQL4 / DQ4	OP4	OP4	-	
DQL5 / DQ5	OP5	OP5	-	
DQL6 / DQ6	OP6	OP6	-	
DQL7 / DQ7	OP7	OP7	-	
DQU0	OP0	-	-	
DQU1	OP1	-	-	
DQU2	OP2	-	-	
DQU3	OP3	-	-	
DQU4	OP5	-	-	
DQU6	OP6	-	-	
DQU7	OP7	-	-	
DML_n / DM_n	INVALID	INVALID	-	
DMU_n	INVALID	-	-	



Flash Memory Summit

This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 152.



It’s not quite as awesome as a “fill block of length n” would have been but that would introduce non-determinism into the protocol

Basically, the biggest advantage is power savings

This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 152.



Error Check Scrub

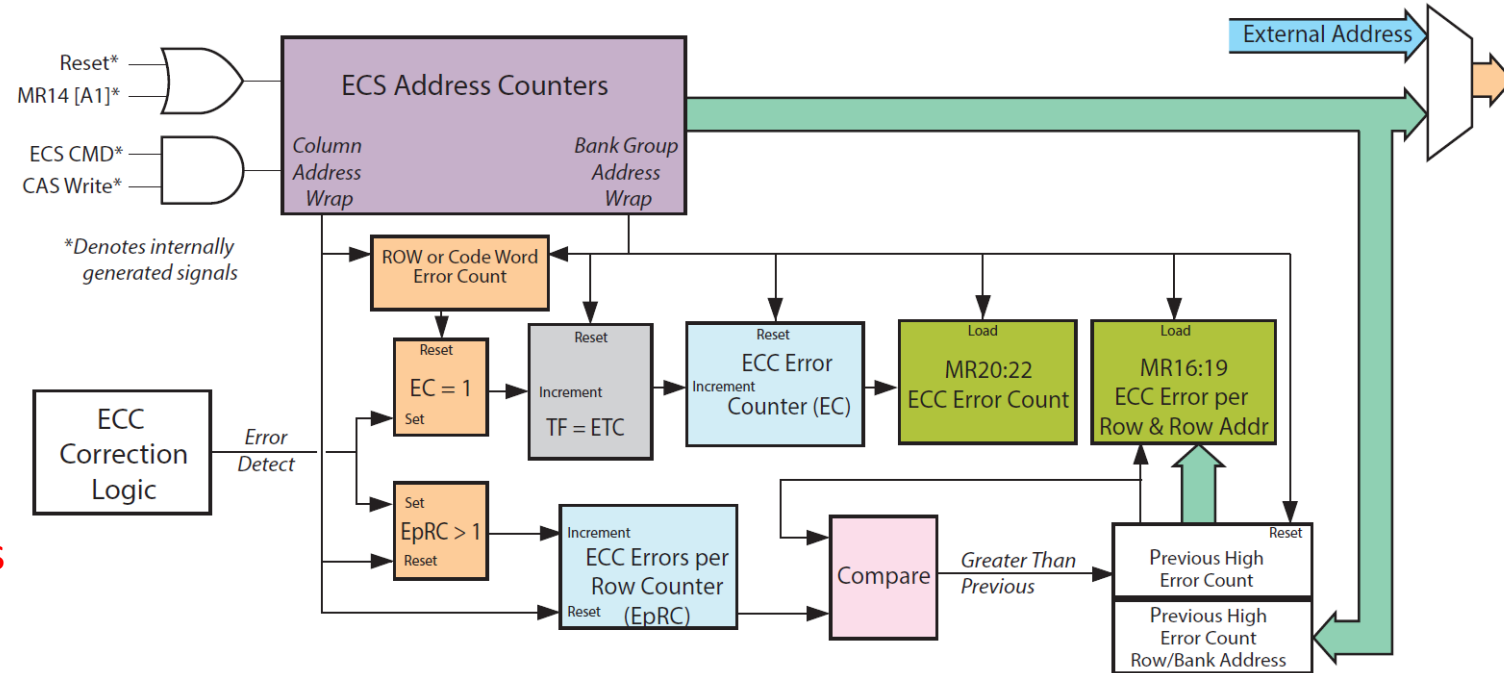
Error Check Scrub (ECS) allows the testing and cleaning of DRAM contents in a bank group

Column-by-column read-modify-write detects and corrects the contents

Automatically increments the column address then the row address, then the bank address

Maintains an error count per row and the address of the row with the most detected errors

This allows the host system to determine when to consider replacing a row of DRAM that is misbehaving too often



This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 154.

Memory Built In Self Test (MBIST)

DDR5 devices run an internal self test when commanded by the Host

Upon completion, the Host polls mode register MR22 to see if additional post package repair is required

OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU					MBIST/mPPR Transparency (optional)		

Function	Type	OP	Description/Data	Notes
MBIST/mPPR Transparency (optional)	R	OP[2:0]	000b: MBIST has not run since INIT or no fails remain after most recent run (default); 001 = Fails remain; 010 = Unrepairable fails remain; 011 = MBIST should be run again; 100-111 = Reserved	1
RFU	RFU	OP[7:3]	RFU	

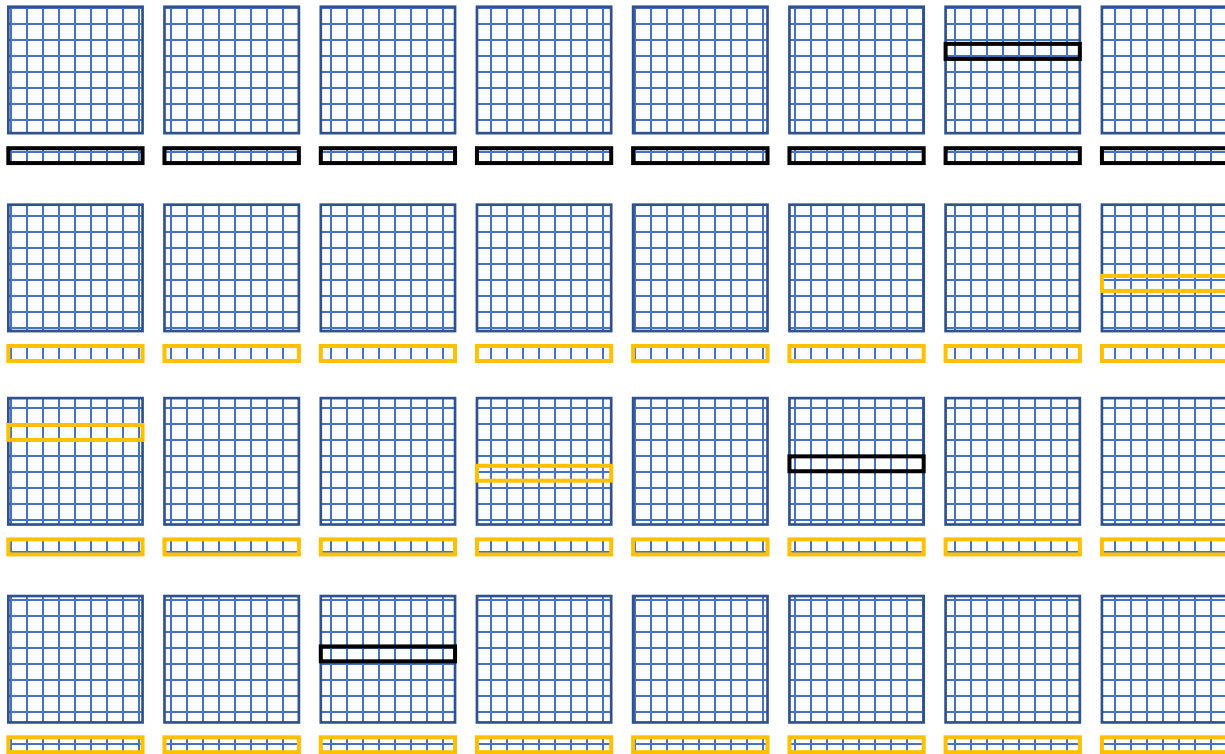
This figure is reproduced, with permission, from JEDEC document JESD79-5B, section 3.5.24.



Redundancy allows post package repair



Bank groups

Banks



32 Total Row Buffers

Vendor specific: one row repair

- Per bank group (required) 
- Per bank (optional) 

Two row repair options

- Soft repair: cleared on reset
- Hard repair: permanent

Works well with Error Check Scrub

- 1) Detect the row with worst error count
- 2) Initiate row repair if it looks too bad



ALERT

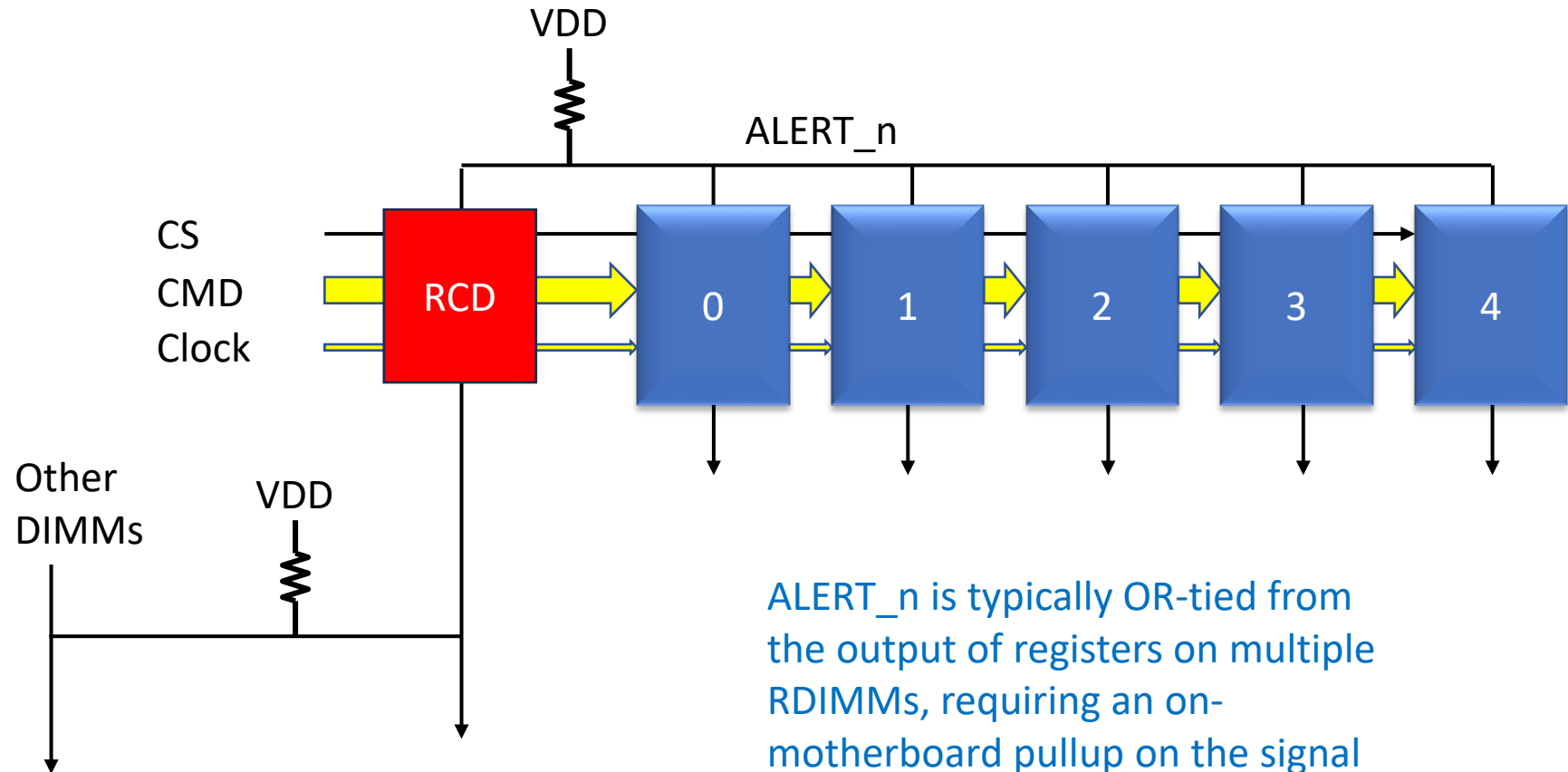
ALERT_n is an OR-tied output from each DRAM – any DRAM can pull it low (active)

Pull-up resistor required on motherboard

Slow: takes multiple clocks

ALERT_n functions:

- Completion of MBIST
- Completion of PPR
- Data CRC failures



ALERT_n is typically OR-tied from the output of registers on multiple RDIMMs, requiring an on-motherboard pullup on the signal

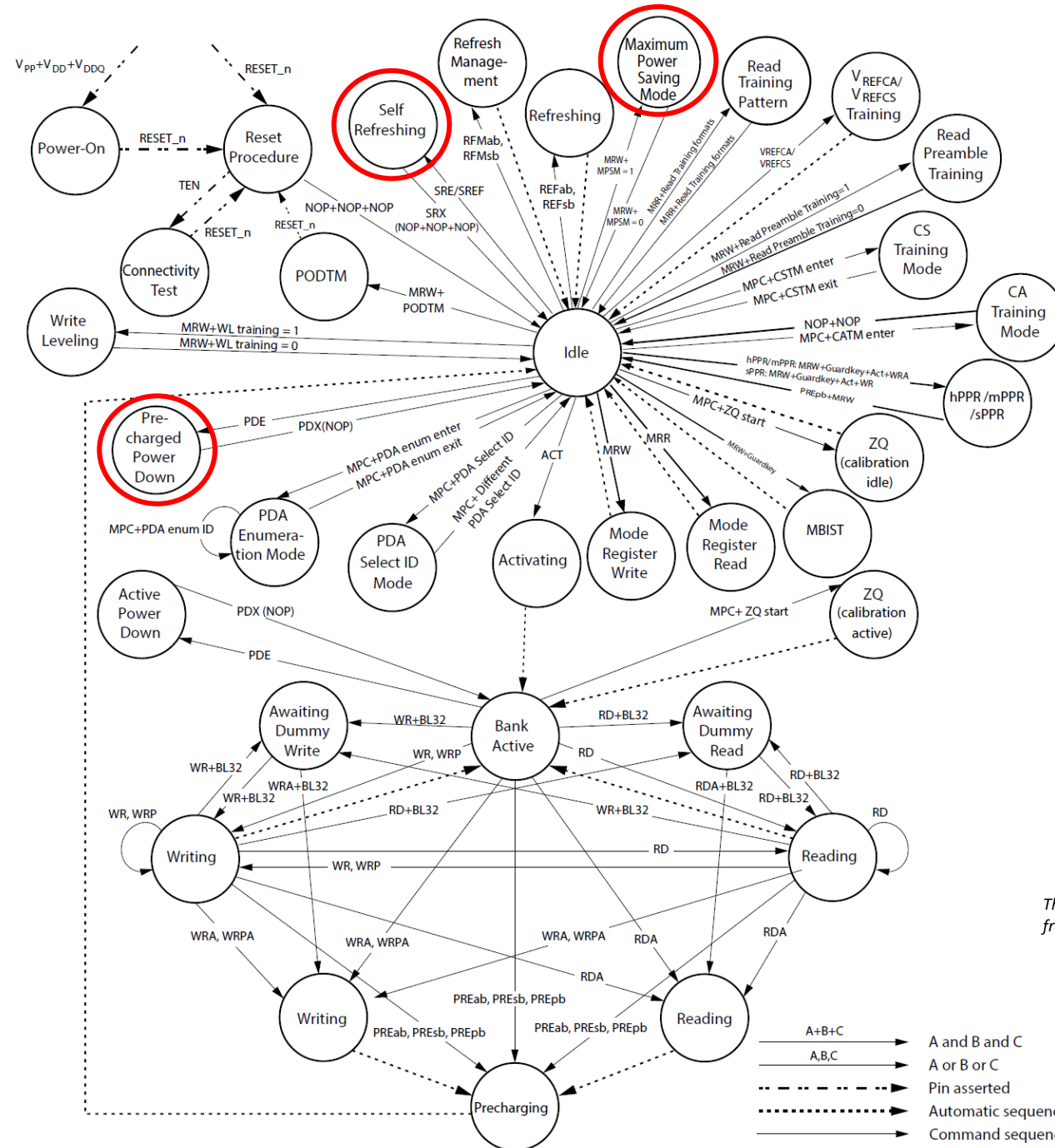
Additional function:

- Address bus errors

Low power modes



- Precharge power down
Clock running, banks closed
Data preserved
- Self Refresh
Power on, clocks stopped
Data preserved
- Maximum Power Savings Mode
Power on
Data NOT preserved
- Power off



*This figure is reproduced, with permission,
from JEDEC document JESD79-5B, section 3.1.*

DDR5 Incorporates an internal thermal sensor

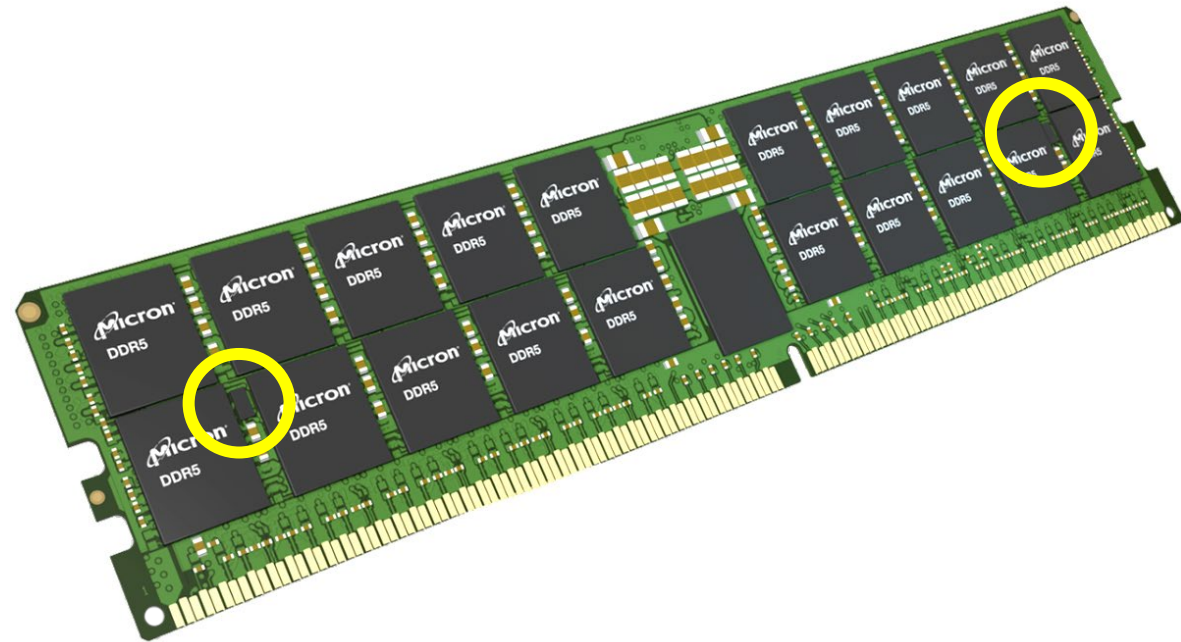
Function	Register Type	Operand	Data
Minimum Refresh Rate	R	OP[2:0]	<p>If Wide Range is not supported (OP[5]=0):</p> <p>000_B: RFU</p> <p>001_B: tREFI x1 (1x Refresh Rate), <80°C nominal</p> <p>010_B: tREFI x1 (1x Refresh Rate), 80-85°C nominal</p> <p>011_B: tREFI /2 (2x Refresh Rate), 85-90°C nominal</p> <p>100_B: tREFI /2 (2x Refresh Rate), 90-95°C nominal</p> <p>101_B: tREFI /2 (2x Refresh Rate), >95°C nominal</p> <p>110_B: RFU</p> <p>111_B: RFU</p> <p>If Wide Range is supported (OP[5]=1):</p> <p>000_B: tREFI x1 (1x Refresh Rate), <75°C nominal</p> <p>001_B: tREFI x1 (1x Refresh Rate), 75-80°C nominal</p> <p>010_B: tREFI x1 (1x Refresh Rate), 80-85°C nominal</p> <p>011_B: tREFI /2 (2x Refresh Rate), 85-90°C nominal</p> <p>100_B: tREFI /2 (2x Refresh Rate), 90-95°C nominal</p> <p>101_B: tREFI /2 (2x Refresh Rate), 95-100°C nominal</p> <p>110_B: tREFI /2 (2x Refresh Rate), >100°C nominal</p> <p>111_B: RFU</p>

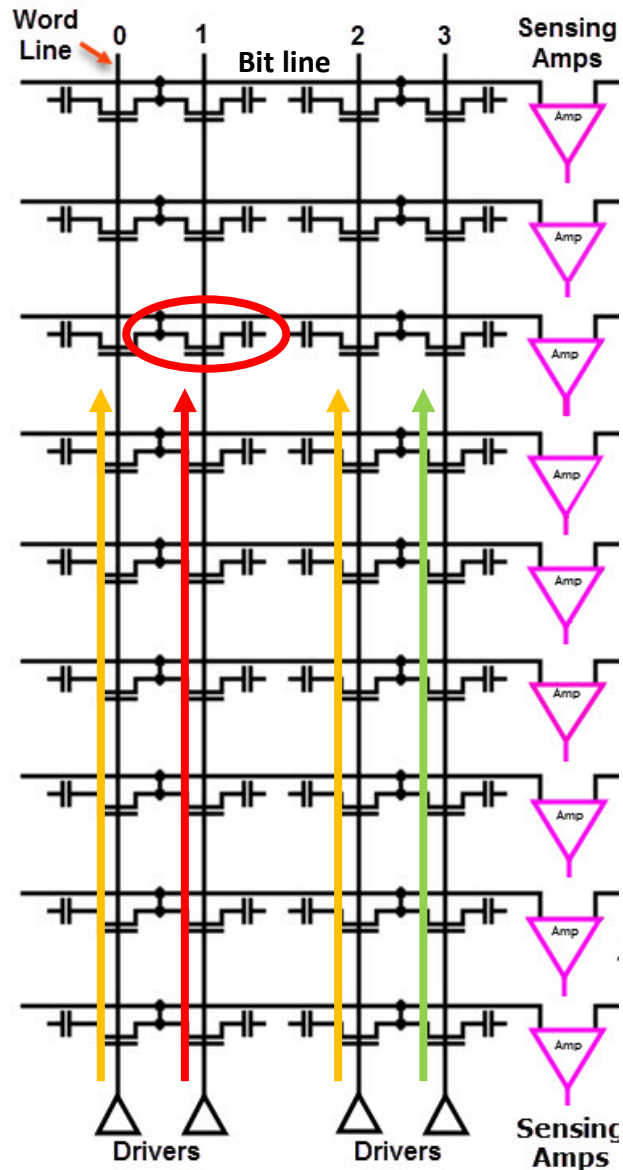
This figure is reproduced, with permission, from JEDEC document JESD79-5B, table 25.

This is a bit redundant is that server memory modules include 2 thermal sensors with 0.5° accuracy, but it's never a bad idea to have more checkpoints

The DRAM has a fairly coarse thermal sensor on-chip

The mode register is driven by the DRAM to hint to the Host controller that it might need a higher refresh rate



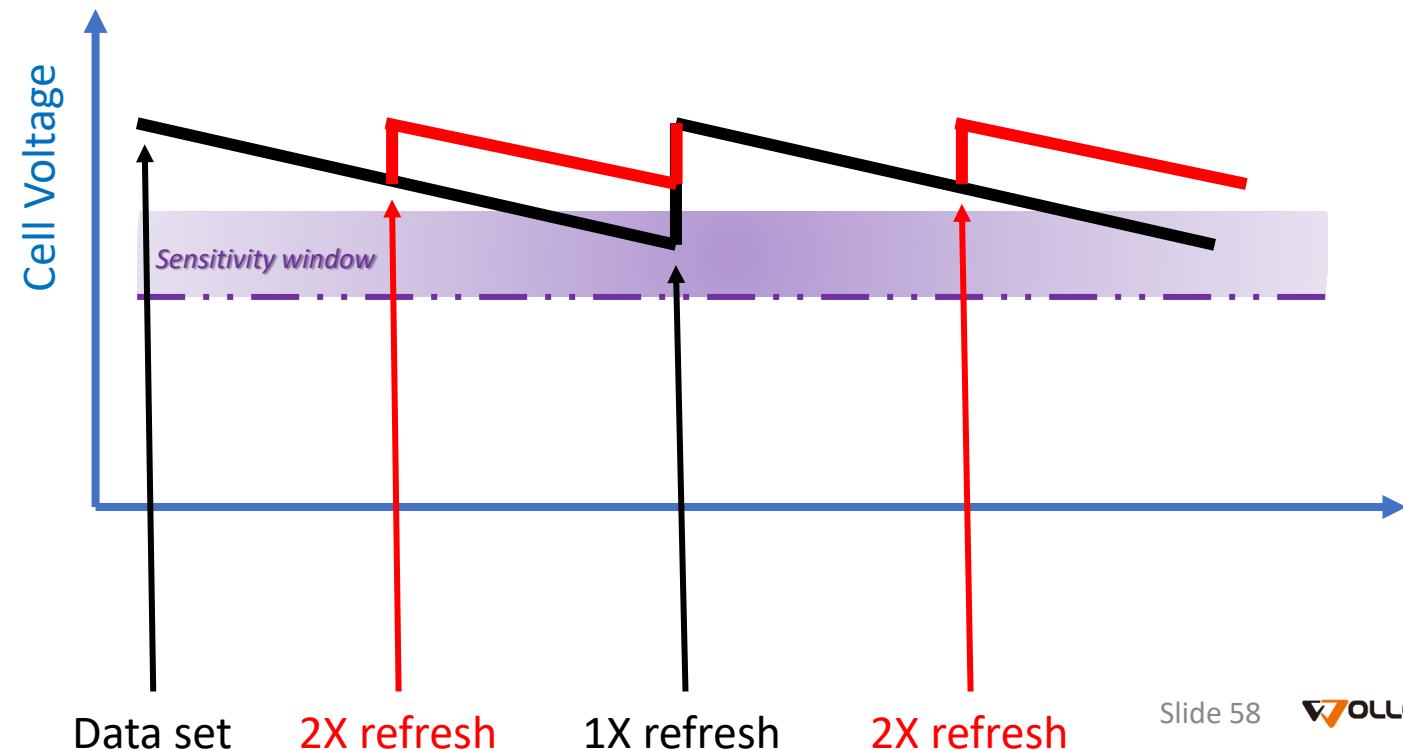


- Activated row
- N±1 effect
- N±2 effect

Refresh Management (RFM)

Some security attacks (such as “rowhammer”) exploit the fundamental crosstalk between long metal lines in order to flip bits, injecting unwanted data changes into critical memory regions

Refresh Management (RFM) techniques avoid the refresh doubling performance penalties by detecting and correcting for attacks





Refresh Management (RFM) Commands

Similar to refresh commands, but also refresh adjacent rows in the “blast radius”

Need triggered by counting activations of a specific DRAM row

Some Refresh Management (RFM) Features

Rolling Accumulated ACT Initial Management Threshold (RAAIMT): DRAM supplier specific suggestion on the number of activations of a given row before action should be taken (mode register readable value)

...but these “suggested RFMs” can be postponed until...

Rolling Accumulated ACT Maximum Management Threshold (RAAMMT): DRAM supplier specific insistence on the maximum number of activations before you’re potentially in deep shit (mode register readable value)

Adaptive Refresh Management (ARFM): Allows varying selectable levels of security with different performance impacts

Directed Fresh Management (DRFM): Recognizes that rows refreshed using RFM commands with “blast radius” don’t also need standard refresh and allows the DRAM to relax subsequent refreshes in that area

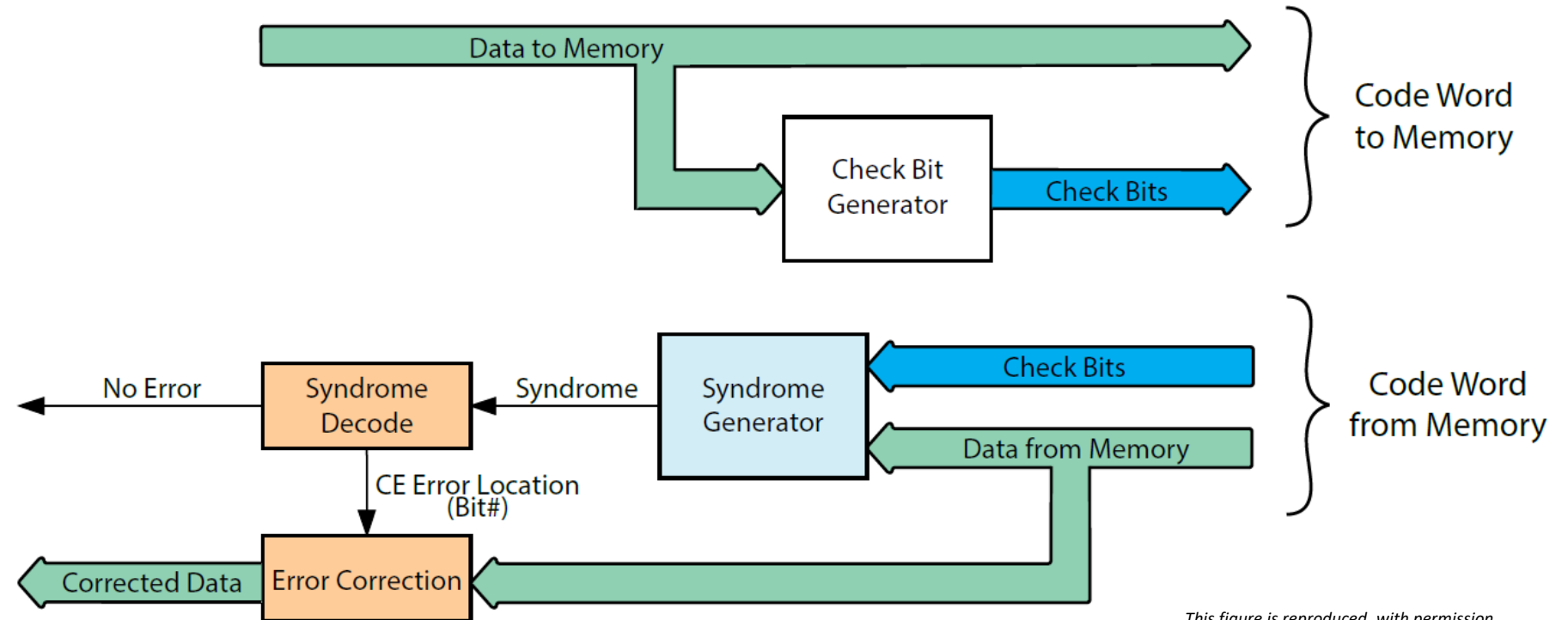


CPU cache lines are almost universally 64 bytes

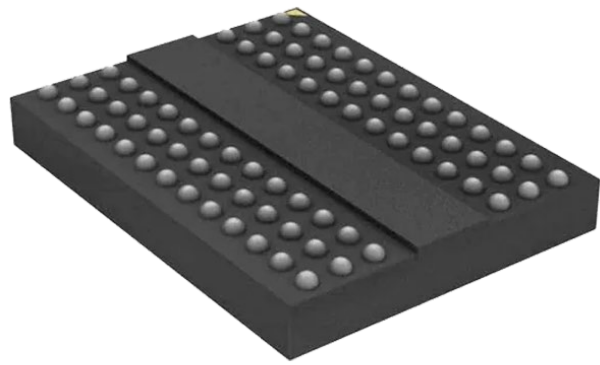
DDR5 ECC is calculated with 128-bit granularity

DRAMs with 8-bit interfaces align well with cache line size

DRAMs with 4-bit interfaces don't, so must merge 64-bit operations with 128-bit internal storage to calculate ECC



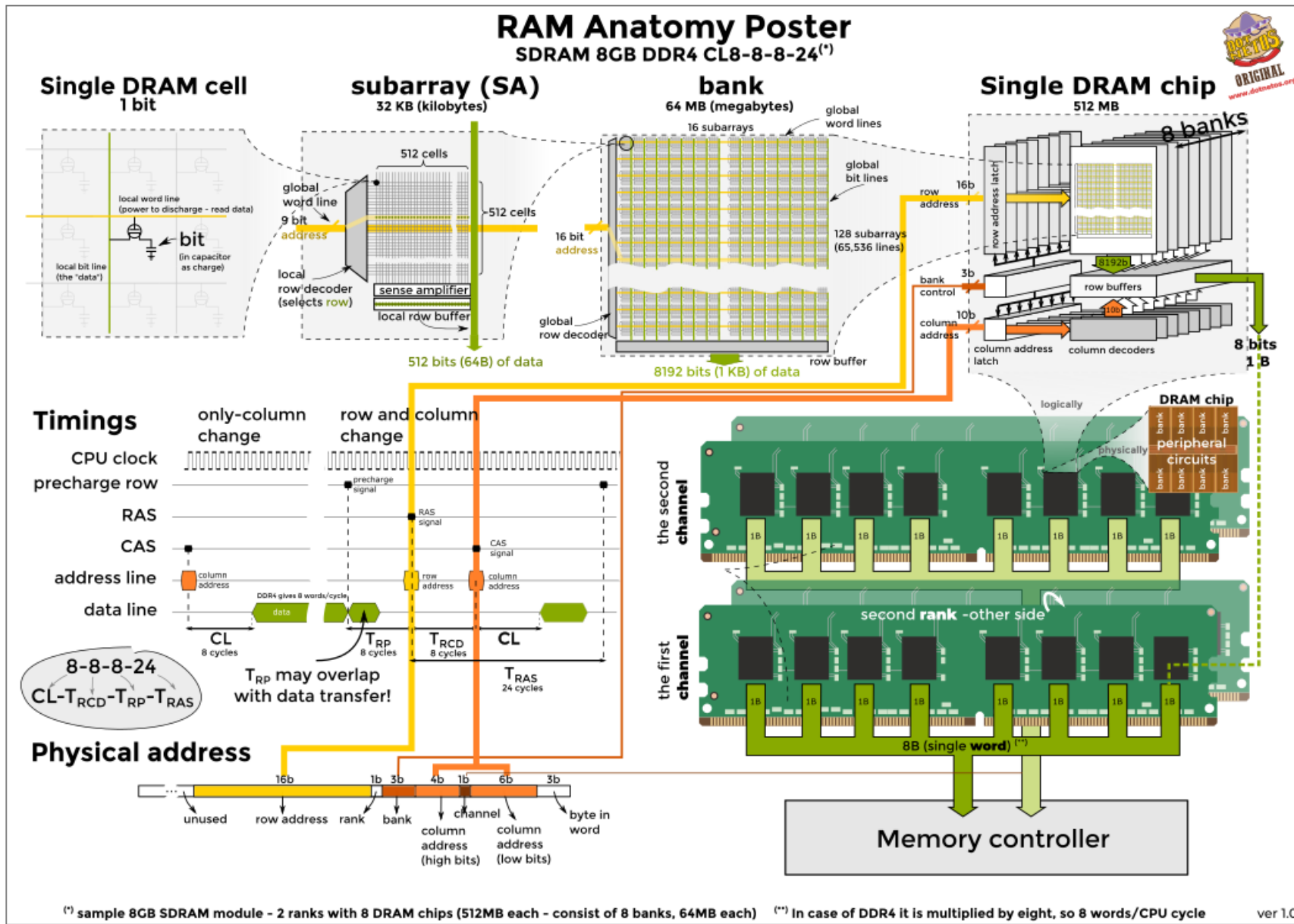
This figure is reproduced, with permission, from JEDEC document JESD79-5B, figure 153.



Switching gears...

Any chip questions before
we move on?





Once we've built a DRAM, we still need to interface it with the rest of the system

Because there is no one standard system, a variety of memory module solutions are defined

DDR5 Memory Modules

Module	Meaning	Markets
SODIMM	Small Outline Dual In-Line Memory Module (DIMM)	Notebooks, telecom
UDIMM	Unbuffered DIMM	Desktop
RDIMM	Registered DIMM	Servers, workstations

LRDIMM? Hmmmm, we should chat about this over a cold beer



Common Features of DDR5 Memory Modules

- SidebandBus system management interface
- Serial Presence Detect (SPD) with SidebandBus Hub
- Programmable Power Management IC (PMIC) for on-module voltage regulation



SidebandBus System Management

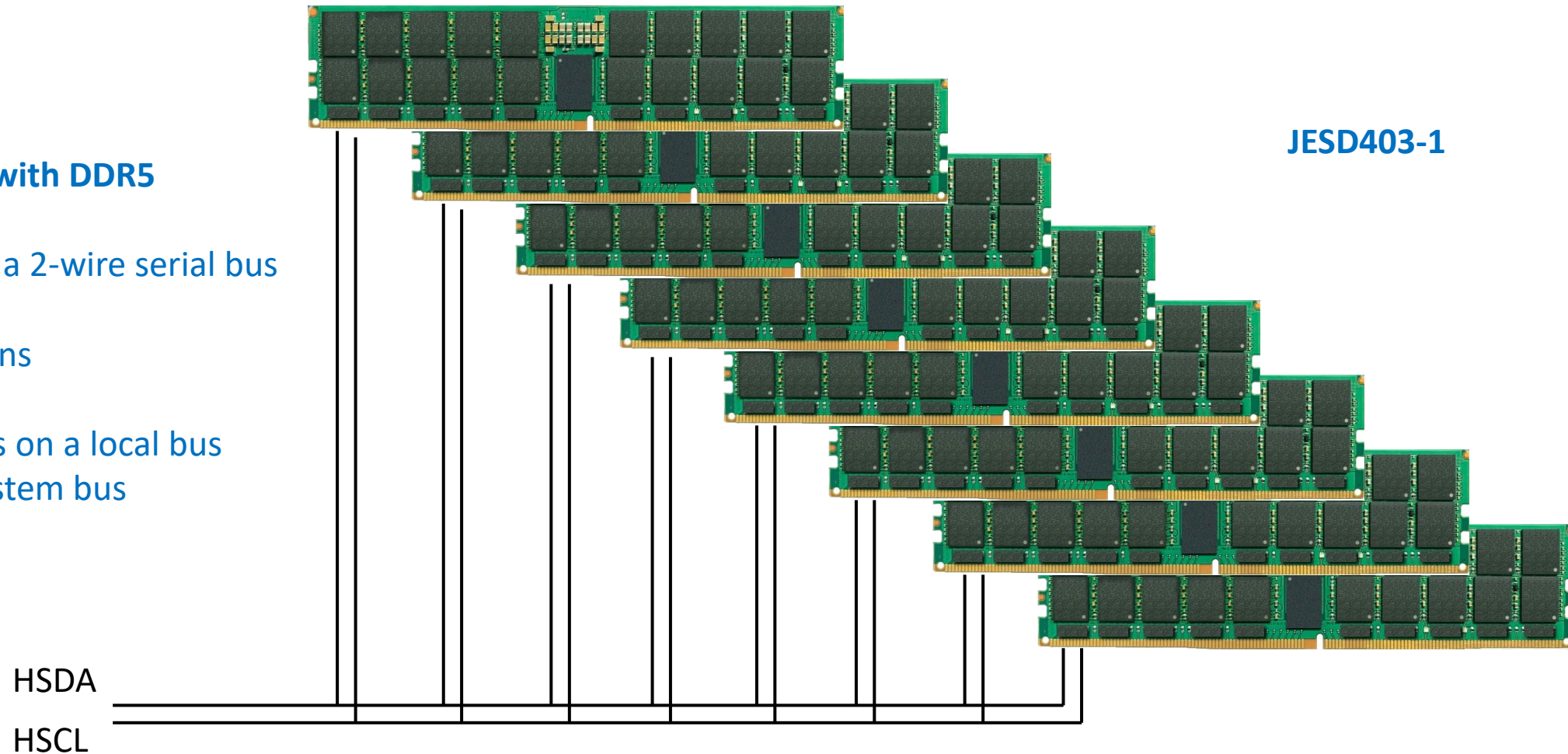
Introduced with DDR5

Up to 8 modules share a 2-wire serial bus

I3C Basic with extensions

Allows multiple devices on a local bus
without loading the system bus

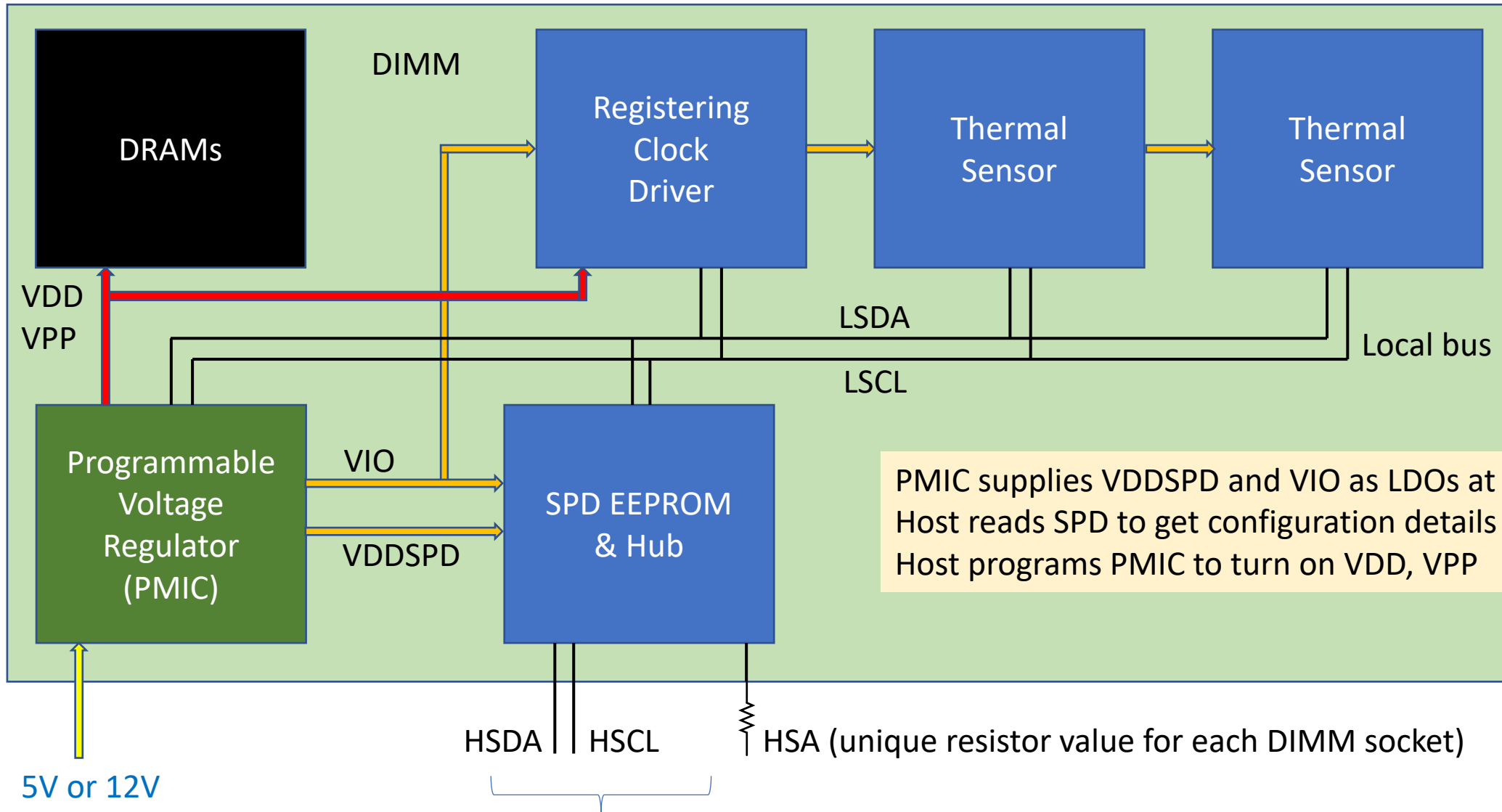
JESD403-1



Module Hub Architecture



Flash Memory Summit



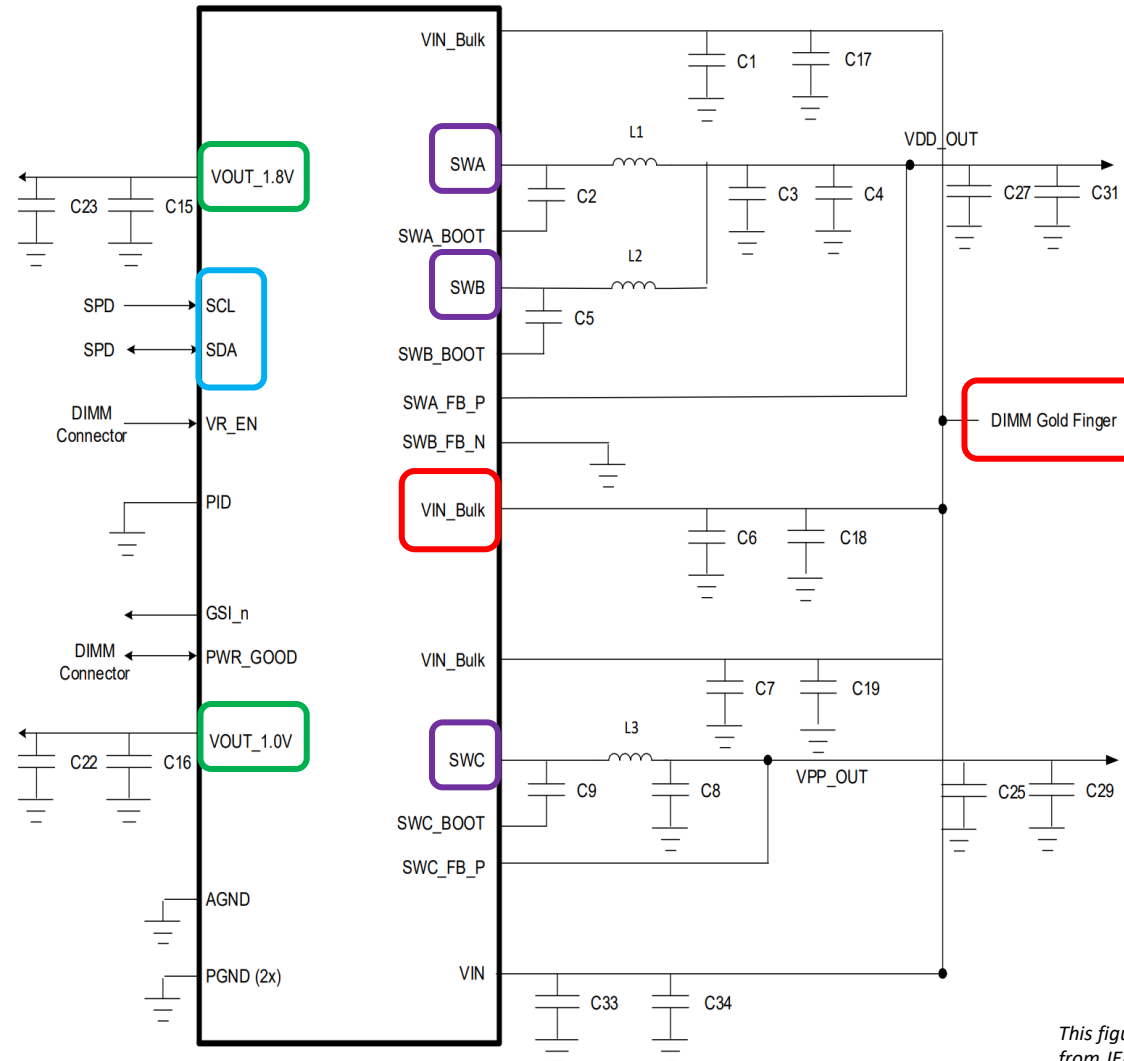
PMIC supplies VDDSPD and VIO as LDOs at power on
Host reads SPD to get configuration details
Host programs PMIC to turn on VDD, VPP

SidebandBus from Host, shared with up to 8 DIMMs total

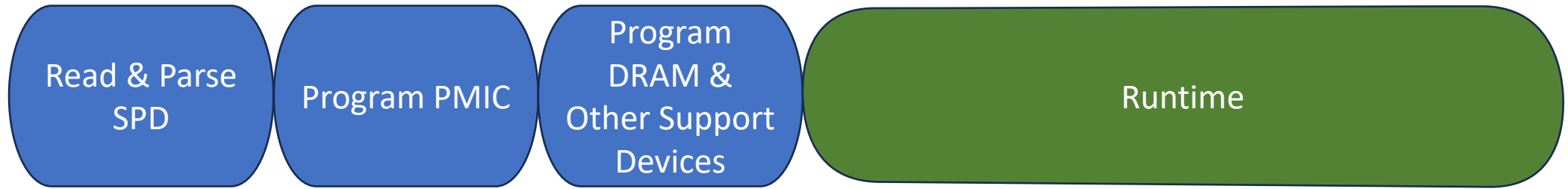
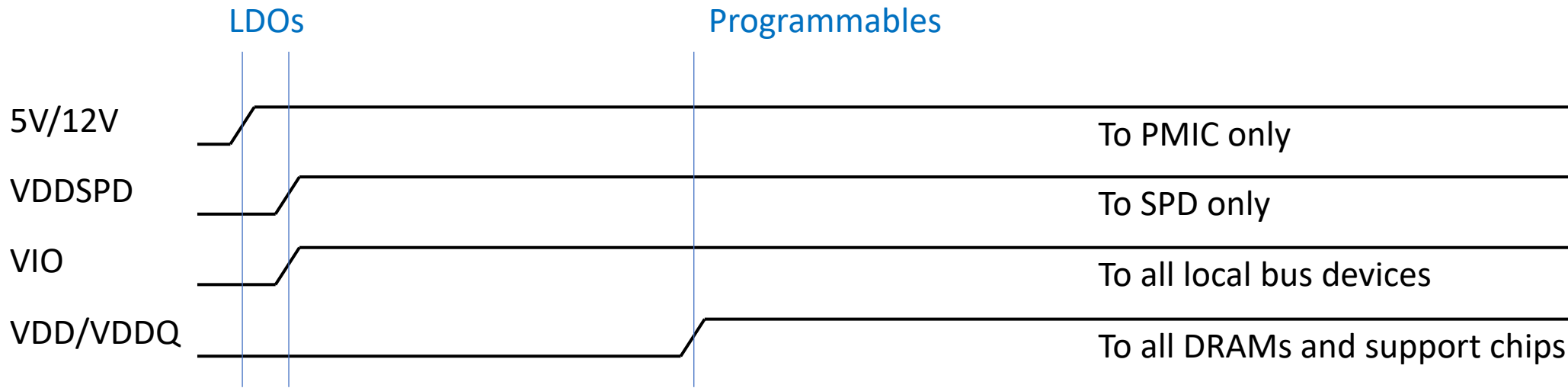


PMIC: DIMM Voltage Regulator

- 5V input for UDIMM, SODIMM
- 12V input for RDIMM
- Low dropout regulators (LDOs) for SPD supply and I/O
- Switchers for the rest of the module supply voltages
- SidebandBus interface for programmability



This figure is reproduced, with permission, from JEDEC document JESD301-1A.02, figure 9.



VDDSPD and VIO being LDOs come up automatically
Host can now read the SPD contents
PMIC raises the VDD/VDDQ to other devices
Host programs the DRAM, RCD, etc. based on SPD contents



During runtime, SidebandBus is open for telemetry gathering or error logging

Read & Parse Base Config

DRAM parameters

0x014	SDRAM Minimum Cycle Time ($t_{CKAVGmin}$), Least Significant Byte
0x015	SDRAM Minimum Cycle Time ($t_{CKAVGmin}$), Most Significant Byte
0x01E	SDRAM Read Command to First Data (t_{AA}), Least Significant Byte
0x01F	SDRAM Read Command to First Data (t_{AA}), Most Significant Byte

This figure is reproduced, with permission,
from JEDEC document JESD400-5, section 7.1.

Bits 3~0
Base Module Type
0000: Reserved
0001: RDIMM
0010: UDIMM
0011: SODIMM
0100: LRDIMM
0101: Reserved
0110: Reserved
0111: Reserved
1000: Reserved
1001: Reserved
1010: DDIMM
1011: Solder down
1100: Reserved
1101: Reserved
1110: Reserved
1111: Reserved

Module type

This figure is reproduced, with permission,
from JEDEC document JESD400-5, section 8.1.4.

Block	Range		Description
0	0~63	0x000~0x03F	Base Configuration and DRAM Parameters
1	64~127	0x040~0x07F	Base Configuration and DRAM Parameters
2	128~191	0x080~0x0BF	Reserved for future use
3	192~239	0x0C0~0x0EF	Common Module Parameters -- See annex A.0 for details
	240~255	0x0D0~0x0FF	Standard Module Parameters -- See annexes A.x for details
4	256~319	0x100~0x13F	Standard Module Parameters -- See annexes A.x for details
5	320~383	0x140~0x17F	Standard Module Parameters -- See annexes A.x for details
6	384~447	0x180~0x1BF	Standard Module Parameters -- See annexes A.x for details
7	448~509	0x1C0~0x1FD	Reserved for future use
	510~511	0x1FE~0x1FF	CRC for SPD bytes 0~509
8	512~575	0x200~0x23F	Manufacturing information
9	576~639	0x240~0x27F	Manufacturing information
10	640~703	0x280~0x2BF	End User Programmable
11	704~767	0x2C0~0x2FF	End User Programmable
12	768~831	0x300~0x33F	End User Programmable
13	832~895	0x340~0x37F	End User Programmable
14	896~959	0x380~0x3BF	End User Programmable
15	960~1023	0x3C0~0x3FF	End User Programmable

This figure is reproduced, with permission,
from JEDEC document JESD400-5, section 3.

SPD Revision for Base Configuration Parameters										
Production Status	SPD Revision	Encoding Level				Additions Level				Hex
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Pre-production	Revision 0.0	0	0	0	0	0	0	0	0	00
	Revision 0.1	0	0	0	0	0	0	0	1	01
	...	-	-	-	-	-	-	-	-	-
	Revision 0.9	0	0	0	0	1	0	0	1	09
Production	Revision 1.0	0	0	0	1	0	0	0	0	10
	Revision 1.1	0	0	0	1	0	0	0	1	11
	...	-	-	-	-	-	-	-	-	...

This figure is reproduced, with permission,
from JEDEC document JESD400-5, section 8.1.2.

SPD Revisions regime allows:

- Old system with new memory module
 - Interpret the subset of SPD that was valid at that time
- New system with old memory module
 - Interpret the subset of SPD that was valid as of that SPD rev
- New system with new memory module
 - Full SPD interpretation to get all the features



Flash Memory Summit



What is “downbinning” and how does it work?

The SPD documents the performance of a DRAM (and support devices) at the maximum frequency of operation

SPD bytes 20-21: SDRAM Minimum Cycle Time (tCKmin) in picoseconds

Timing parameters such as access time (tAA), precharge time (tRP), etc are expressed in picoseconds

SPD bytes 30-31 (tAA), 34-35 (tRP), etc.

Translating these timing parameters from picoseconds to clocks to program the memory controller is dependent on the application operating frequency

While technically any operating frequency should work, suppliers only guarantee operation at standard data rates

DDR5-3200, DDR5-3600, DDR5-4000 ... DDR5-8800

To avoid off-by-one errors, a standard rounding algorithm is applied, e.g.

$tRP \text{ in ps} \div tCK_{\text{application in ps}} \rightarrow tRP \text{ in clocks at the application frequency}$

However, due to the inherent inaccuracy of digital math, performance may be lost with this simplistic algorithm

To avoid performance loss, the algorithm is modified with a 0.3% correction factor



Downbinning the CAS Latency is more complicated

Speed Bin				DDR5-3200AN		DDR5-3200B		DDR5-3200BN		DDR5-3200C		Unit		
CL-nRCD-nRP				24-24-24		26-26-26		26-26-26		28-28-28				
Parameter			Symbol	min	max	min	max	min	max	min	max			
Read command to first data			tAA	15.000	22.222	16.250	22.222	16.250	22.222	17.500	22.222	ns		
Activate to Read or Write command delay time			tRCD	15.000	-	16.250	-	16.250	-	17.500	-	ns		
Row Precharge time			tRP	15.000	-	16.250	-	16.250	-	17.500	-	ns		
Activate to Precharge command period			tRAS	32.000	5 * tREFI1 (Norm) 9 * tREFI2 (FGR)	32.000	5 * tREFI1 (Norm) 9 * tREFI2 (FGR)	32.000	5 * tREFI1 (Norm) 9 * tREFI2 (FGR)	32.000	5 * tREFI1 (Norm) 9 * tREFI2 (FGR)	ns		
Activate to Activate or Refresh command period			tRC (tRAS +tRP)	47.000	-	48.250	-	48.250	-	49.500	-	ns		
CAS Write Latency			CWL	CL-2								nCK		
Speed Bin ⁵	tAAmin (ns) ⁵	tRCDmin tRPmin (ns) ⁵	Read CL ¹²	Supported Frequency Down Bins										
-	20.952	-	22	tCK(AVG)	0.952	1.010	0.952	1.010	0.952	1.010	0.952	1.010	ns	
3200C	17.500	17.500	28	tCK(AVG)	0.625	0.681	0.625	0.681	0.625	0.681	0.625	0.681	ns	
3200BN,B	16.250	16.250	26	tCK(AVG)	0.625	0.681	0.625	0.681	0.625	0.681	RESERVED		ns	
3200AN	15.000	15.000	24	tCK(AVG)	0.625	0.681	RESERVED							ns
Supported CL				22,24,26,28		22,26,28		22,26,28		22,28		nCK		

This figure is reproduced, with permission, from JEDEC document JESD79-5B, table 273.

						SPD[24]							
Mono Speed Bin	SPD[24]	SPD[25]	SPD[26]	SPD[27]	SPD[28]	20	22	24	26	28	30	32	34
3200AN	0x1E	0x00	0x00	0x00	0x00	0	1	1	1	1	0	0	0
3200B	0x1A	0x00	0x00	0x00	0x00	0	1	0	1	1	0	0	0
3200BN	0x1A	0x00	0x00	0x00	0x00	0	1	0	1	1	0	0	0
3200C	0x12	0x00	0x00	0x00	0x00	0	1	0	0	1	0	0	0
3200AN	Next valid CL					22	22	24	26	28	0	0	0
3200B						22	22	26	26	28	0	0	0
3200BN						22	22	26	26	28	0	0	0
3200C						22	22	28	28	28	0	0	0

Each DDR5 speed bin may support a different set of CAS latencies (tAA ÷ tCK)

After the calculation of CAS latency via the rounding algorithm, the BIOS needs to round up to the next supported CAS latency

The CL masks are also provided in the SPD (bytes 24-28)



Read & Parse
Common
Module Data

(Common): SPD Revision for Module Information
Byte 192 (0x0C0)

Module revision is distinct from SPD
base revision, minimizing BIOS work
when things change

Block	Range		Description
0	0~63	0x000~0x03F	Base Configuration and DRAM Parameters
1	64~127	0x040~0x07F	Base Configuration and DRAM Parameters
2	128~191	0x080~0x0BF	Reserved for future use
3	192~239	0x0C0~0x0EF	Common Module Parameters -- See annex A.0 for details
	240~255	0x0D0~0x0FF	Standard Module Parameters -- See annexes A.x for details
4	256~319	0x100~0x13F	Standard Module Parameters -- See annexes A.x for details
5	320~383	0x140~0x17F	Standard Module Parameters -- See annexes A.x for details
6	384~447	0x180~0x1BF	Standard Module Parameters -- See annexes A.x for details
7	448~509	0x1C0~0x1FD	Reserved for future use
	510~511	0x1FE~0x1FF	CRC for SPD bytes 0~509
8	512~575	0x200~0x23F	Manufacturing information
9	576~639	0x240~0x27F	Manufacturing information
10	640~703	0x280~0x2BF	End User Programmable
11	704~767	0x2C0~0x2FF	End User Programmable
12	768~831	0x300~0x33F	End User Programmable
13	832~895	0x340~0x37F	End User Programmable
14	896~959	0x380~0x3BF	End User Programmable
15	960~1023	0x3C0~0x3FF	End User Programmable

SPD does not have a code for
module capacity; this is
calculated from the module
configuration

Capacity in bytes =
Number of sub-channels per DIMM *
Primary bus width per sub-channel / SDRAM I/O Width *
Die per package *
SDRAM density per die / 8 *
Package ranks per sub-channel

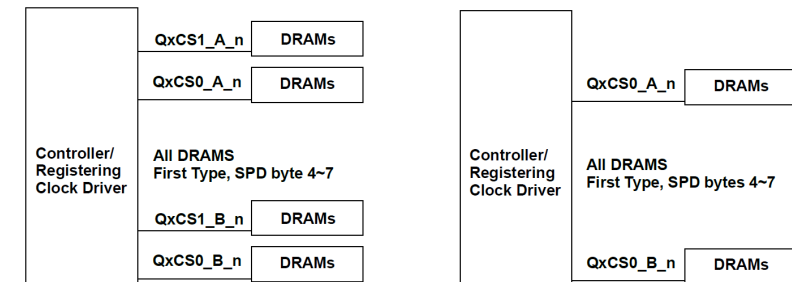
Device Types			
	Bit 7	Bit 6	Bits 5 ~ 4
Device	Devices Installed		Reserved
SPD	0 = Not installed 1 = Installed	Reserved; must be coded as 0	0000: SPD5118 (see JESD300-5) 0001: ESPD5216 (see JESD316-5) All other codes reserved
PMIC 0	0 = Not installed 1 = Installed		0000: PMIC5000 (see JESD301-1) 0001: PMIC5010 (see JESD301-1) 0010: PMIC5100 (see JESD301-2) All other codes reserved
PMIC 1	0 = Not installed 1 = Installed		
PMIC 2	0 = Not installed 1 = Installed		
Thermal Sensors	0 = TS0 Not installed 1 = TS0 Installed	0 = TS1 Not installed 1 = TS1 Installed	0000: TS5111 (see JESD302-1) 0001: TS5110 (see JESD302-1) All other codes reserved

Devices common to
all module types

Including solder
down

*This figure is reproduced, with permission,
from JEDEC document JESD400-5, section A.0.*

Configuration of
DRAMs into channels
and subchannels



*This figure is reproduced, with permission,
from JEDEC document JESD400-5, section A.0.*



(Common): SPD Revision for Module Information Byte 192 (0x0C0)

Common section: Module SPD revision

Base section: Module type

Bits 3~0
Base Module Type
0000: Reserved
0001: RDIMM
0010: UDIMM
0011: SODIMM
0100: LRDIMM
0101: Reserved
0110: Reserved
0111: Reserved
1000: Reserved
1001: Reserved
1010: DDIMM
1011: Solder down
1100: Reserved
1101: Reserved
1110: Reserved
1111: Reserved

Devices unique to the module type

Module Specific Device Types				
	Bit 7	Bit 6	Bits 5 ~ 4	Bits 3 ~ 0
Device	Devices Installed		Reserved	Device Type
Registering Clock Driver (RCD)	0 = Not installed 1 = Installed	Reserved; must be coded as 0	Reserved. Must be coded as 00	0000: DDR5RCD01 (see JESD82-511) 0001: DDR5RCD02 (see JESD82-512) 0010: DDR5RCD03 (see JESD82-513) 0011: DDR5RCD04 (see JESD82-514) All other codes reserved
Data Buffers (DB)	0 = Not installed (RDIMM) 1 = Installed (LRDIMM)			0000: DDR5DB01 (see JESD82-521) 0001: DDR5DB02 (see JESD82-522) All other codes reserved RDIMM: code as 0000

*This figure is reproduced, with permission,
from JEDEC document JESD400-5, section A.3.*

Read & Parse
Module
Specific Data

Optimal driver settings for this vendor's implementation

CKD-RW02: QCK Driver Characteristics			
Bits 7 ~ 6	Bits 5 ~ 4	Bits 3 ~ 2	Bits 1 ~ 0
CHB QCK_1_B_t/QCK_1_B_c	CHB QCK_0_B_t/QCK_0_B_c	CHA QCK_1_A_t/QCK_1_A_c	CHA QCK_0_A_t/QCK_0_A_c
00 = Light Drive 01 = Moderate Drive 10 = Strong Drive 11 = Weak Drive	00 = Light Drive 01 = Moderate Drive 10 = Strong Drive 11 = Weak Drive	00 = Light Drive 01 = Moderate Drive 10 = Strong Drive 11 = Weak Drive	00 = Light Drive 01 = Moderate Drive 10 = Strong Drive 11 = Weak Drive

*This figure is reproduced, with permission,
from JEDEC document JESD400-5, section A.3.*

Block	Range		Description
0	0~63	0x000~0x03F	Base Configuration and DRAM Parameters
1	64~127	0x040~0x07F	Base Configuration and DRAM Parameters
2	128~191	0x080~0x0BF	Reserved for future use
	192~239	0x0C0~0x0FF	Common Module Parameters -- See annex A.0 for details
3	240~255	0x0D0~0x0FF	Standard Module Parameters -- See annexes A.x for details
4	256~319	0x100~0x13F	Standard Module Parameters -- See annexes A.x for details
5	320~383	0x140~0x17F	Standard Module Parameters -- See annexes A.x for details
6	384~447	0x180~0x1BF	Standard Module Parameters -- See annexes A.x for details
	448~509	0x1C0~0x1FD	Reserved for future use
7	510~511	0x1FE~0x1FF	CRC for SPD bytes 0~509
8	512~575	0x200~0x23F	Manufacturing information
9	576~639	0x240~0x27F	Manufacturing information
10	640~703	0x280~0x2BF	End User Programmable
11	704~767	0x2C0~0x2FF	End User Programmable
12	768~831	0x300~0x33F	End User Programmable
13	832~895	0x340~0x37F	End User Programmable
14	896~959	0x380~0x3BF	End User Programmable
15	960~1023	0x3C0~0x3FF	End User Programmable

Optional in the User Area: standard error logging



This tracks module issues even if the module is moved to another system

Block	Range		Description
0	0~63	0x000~0x03F	Base Configuration and DRAM Parameters
1	64~127	0x040~0x07F	Base Configuration and DRAM Parameters
2	128~191	0x080~0x0BF	Reserved for future use
3	192~239	0x0C0~0x0EF	Common Module Parameters -- See annex A.0 for details
	240~255	0x0D0~0x0FF	Standard Module Parameters -- See annexes A.x for details
	256~319	0x100~0x13F	Standard Module Parameters -- See annexes A.x for details
4	320~383	0x140~0x17F	Standard Module Parameters -- See annexes A.x for details
5	384~447	0x180~0x1BF	Standard Module Parameters -- See annexes A.x for details
7	448~509	0x1C0~0x1FD	Reserved for future use
	510~511	0x1FE~0x1FF	CRC for SPD bytes 0~509
8	512~575	0x200~0x23F	Manufacturing information
9	576~639	0x240~0x27F	Manufacturing information
10	640~703	0x280~0x2BF	End User Programmable
11	704~767	0x2C0~0x2FF	End User Programmable
12	768~831	0x300~0x33F	End User Programmable
13	832~895	0x340~0x37F	End User Programmable
14	896~959	0x380~0x3BF	End User Programmable
15	960~1023	0x3C0~0x3FF	End User Programmable

End user data is not write protected

Bytes	Field	Meaning
n ~ n+3	Anchor	Anchor String to identify the beginning of an error log
n + 4	Header	Error Type
n+5 ~ n+13	Address	Error Location
n+14~n+17	Location	Timestamp
n+18	Refresh	Highest DRAM Refresh Settings on the Module
n+19 ~ n+20	Temperature	Module Measured Temperature
n+21 ~ n+23	Reserved	Reserved for future use

This figure is reproduced, with permission, from JEDEC document JESD400-5, section 19.1.

This figure is reproduced, with permission, from JEDEC document JESD400-5, section 19.1.

Error Location								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n+5	Reserved	Reserved	CPU2	CPU1	CPU0	Reserved	CPUMC3	CPUMC2
n+6	CPUMC1	CPUMC0	Reserved	DIMM	CS0_A_n	CS1_A_n	CS0_B_n	CS1_B_n
n+7	Reserved	PAR	CID3/R17	CID2	CID1	CID0	BG2	BG1
n+8	BG0	BA1	BA0	R16	R15	R14	R13	R12
n+9	R11	R10	R9	R8	R7	R6	R5	R4
n+10	R3	R2	R1	R0	C10	C9	C8	C7
n+11	C6	C5	C4	C3	DQS9A_n	DQS8A_n	DQS7A_n	DQS6A_n
n+12	DQS54_n	DQS4A_n	DQS3A_n	DQS2A_n	DQS1A_n	DQS0A_n	DQS9B_n	DQS8B_n
n+13	DQS7B_n	DQS6B_n	DQS54_n	DQS4B_n	DQS3B_n	DQS2B_n	DQS1B_n	DQS0B_n

Timestamp								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n+14	Year						Month MSb	
n+15	Month LSb		Day				Hour MSb	
n+16	Hour LSb				Minute MSb			
n+17	Minute LSb		Second					

This figure is reproduced, with permission, from JEDEC document JESD400-5, section 19.1.



	3200AN Mono	3200B Mono	3200BN Mono	3200C Mono
CL Algorithm	24	26	26	28
CL in nCK	24	26	26	28
RCD in nCK	24	26	26	28
RP in nCK	24	26	26	28
CL per spec	24	26	26	28
nRCD per spec	24	26	26	28
nRP per spec	24	26	26	28
CL pre-mask	24	26	26	28

3600AN Mono	3600B Mono	3600BN Mono	3600C Mono
26	30	30	32
26	30	30	32
26	30	30	32
26	30	30	32
26	30	30	32
26	30	30	32
26	30	30	32
26	30	30	32

4000AN Mono	4000B Mono	4000BN Mono	4000C Mono
28	32	32	36
28	32	32	36
28	32	32	35
28	32	32	35
28	32	32	36
28	32	32	35
28	32	32	35
28	32	32	35

	3200AN Mono	3200B Mono	3200BN Mono	3200C Mono
tCKmin	0.625	0.625	0.625	0.625
tCKmax	1.010	1.010	1.010	1.010
tAA	15.000	16.250	16.250	17.500
tRCD	15.000	16.250	16.250	17.500
tRP	15.000	16.250	16.250	17.500
tRAS	32.000	32.000	32.000	32.000
tRC	47.000	48.250	48.250	49.500
tWR	30.000	30.000	30.000	30.000

3600AN Mono	3600B Mono	3600BN Mono	3600C Mono
0.555	0.555	0.555	0.555
1.010	1.010	1.010	1.010
14.444	16.250	16.666	17.500
14.444	16.250	16.666	17.500
14.444	16.250	16.666	17.500
32.000	32.000	32.000	32.000
46.444	48.250	48.666	49.500
30.000	30.000	30.000	30.000

4000AN Mono	4000B Mono	4000BN Mono	4000C Mono
0.500	0.500	0.500	0.500
1.010	1.010	1.010	1.010
14.000	16.000	16.000	17.500
14.000	16.000	16.000	17.500
14.000	16.000	16.000	17.500
32.000	32.000	32.000	32.000
46.000	48.000	48.000	49.500
30.000	30.000	30.000	30.000

SPD #	tCKmin (ps)	625	625	625	625
20	tCKmin low	0x71	0x71	0x71	0x71
21	tCKmin high	0x02	0x02	0x02	0x02

555	555	555	555
0x2B	0x2B	0x2B	0x2B
0x02	0x02	0x02	0x02

500	500	500	500
0xF4	0xF4	0xF4	0xF4
0x01	0x01	0x01	0x01

SPD #	tCKmax (ps)	1010	1010	1010	1010
22	tCKmax low	0xF2	0xF2	0xF2	0xF2
23	tCKmax high	0x03	0x03	0x03	0x03

1010	1010	1010	1010
0xF2	0xF2	0xF2	0xF2
0x03	0x03	0x03	0x03

1010	1010	1010	1010
0xF2	0xF2	0xF2	0xF2
0x03	0x03	0x03	0x03

SPD #	tAA (ps)	15000	16250	16250	17500
30	tAA low	0x98	0x7A	0x7A	0x5C
31	tAA high	0x3A	0x3F	0x3F	0x44

14444	16250	16666	17500
0x6C	0x7A	0x1A	0x5C
0x38	0x3F	0x41	0x44

14000	16000	16000	17500
0xB0	0x80	0x80	0x5C
0x36	0x3E	0x3E	0x44

SPD #	tRCD (ps)	15000	16250	16250	17500
32	tRCD low	0x98	0x7A	0x7A	0x5C
33	tRCD high	0x3A	0x3F	0x3F	0x44

14444	16250	16666	17500
0x6C	0x7A	0x1A	0x5C
0x38	0x3F	0x41	0x44

14000	16000	16000	17500
0xB0	0x80	0x80	0x5C
0x36	0x3E	0x3E	0x44

SPD #	tRP (ps)	15000	16250	16250	17500
34	tRP low	0x98	0x7A	0x7A	0x5C
35	tRP high	0x3A	0x3F	0x3F	0x44

14444	16250	16666	17500
0x6C	0x7A	0x1A	0x5C
0x38	0x3F	0x41	0x44

14000	16000	16000	17500
0xB0	0x80	0x80	0x5C
0x36	0x3E	0x3E	0x44

SPD #	tRAS (ps)	32000	32000	32000	32000
36	tRAS low	0x00	0x00	0x00	0x00
37	tRAS high	0x7D	0x7D	0x7D	0x7D

32000	32000	32000	32000
0x00	0x00	0x00	0x00
0x7D	0x7D	0x7D	0x7D

32000	32000	32000	32000
0x00	0x00	0x00	0x00
0x7D	0x7D	0x7D	0x7D

SPD #	tRC (ps)	47000	48250	48250	49500
38	tRC low	0x98	0x7A	0x7A	0x5C
39	tRC high	0xB7	0xBC	0xBC	0xC1

46444	48250	48666	49500
0x6C	0x7A	0x1A	0x5C
0xB5	0xBC	0xBE	0xC1

46000	48000	48000	49500
0xB0	0x80	0x80	0x5C
0xB3	0xBB	0xBB	0xC1

SPD #	tWR (ps)	30000	30000	30000	30000
40	tWR low	0x30	0x30	0x30	0x30
41	tWR high	0x75	0x75	0x75	0x75

30000	30000	30000	30000
0x30	0x30	0x30	0x30
0x75	0x75	0x75	0x75

30000	30000	30000	30000
0x30	0x30	0x30	0x30
0x75	0x75	0x75	0x75

	3200AN Mono	3200B Mono	3200BN Mono	3200C Mono
tRAS in nCK	52	52	52	52
tRC in nCK	75	77	77	79
tWR in nCK	48	48	48	48

3600AN Mono	3600B Mono	3600BN Mono	3600C Mono
58	58	58	58
84	87	88	89
54	54	54	54

4000AN Mono	4000B Mono	4000BN Mono	4000C Mono
64	64	64	64
92	96	96	99
60	60	60	60

SPD Calculation Spreadsheet

Pre-calculates the hexadecimal codes needed for all DDR5 speed bins, monolithic, DDP, and 3DS

These values are great for calculating the impact of downbinning (running a fast memory slower than the maximum frequency)

This is only posted and maintained on the JEDEC member's page, however it is allowed to be shared with customers

Ask for it from your supplier if you need it!

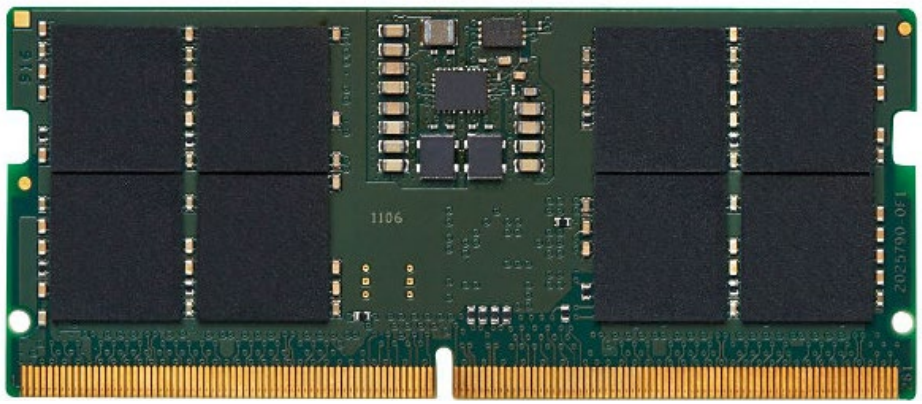
Module routing rules

Reference planes

SODIMM: Small Outline Dual In-Line Memory Module

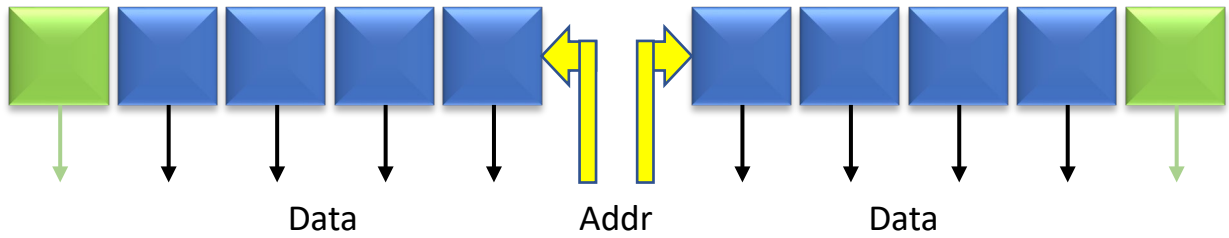


Flash Memory Summit



30.0 mm

69.6 mm



JESD309 common specification

- 5V PMIC supply voltage
- Two 32- or 36-bit subchannels
- x8 or x16 SDRAMs (half of x8 ignored on ECC lane)
- Single data rate address/command bus
- One or two ranks supported (with DDP)

• Right-angle sockets for low profile

JESD309	DDR5 Small Outline Dual Inline Memory Module (SODIMM) Common Specification	
JESD309-S0-RCA	DDR5 Small Outline Dual Inline Memory Module (SODIMM) Raw Card A Annex	1Rx8, 2 channel x 32b/ch
JESD309-S0-RCB	DDR5 Small Outline Dual Inline Memory Module (SODIMM) Raw Card B Annex	2Rx8, 2 channel x 32b/ch
JESD309-S0-RCC	DDR5 Small Outline Dual Inline Memory Module (SODIMM) Raw Card C Annex	1Rx16, 2 channel x 32b/ch
JESD309-S4-RCD	DDR5 Small Outline Dual Inline Memory Module with 4-bit ECC (EC4 SODIMM) Raw Card D Annex	1Rx8 ECC, 2 channel x 36b/ch
JESD309-S4-RCE	DDR5 Small Outline Dual Inline Memory Module with 4-bit ECC (EC4 SODIMM) Raw Card E Annex	2Rx8 ECC, 2 channel x 36b/ch

UDIMM: Unbuffered Dual In-Line Memory Module

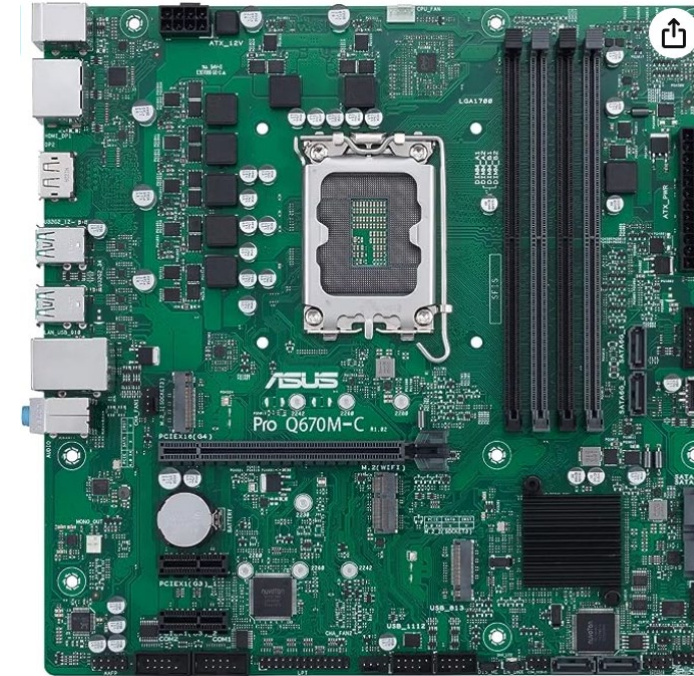
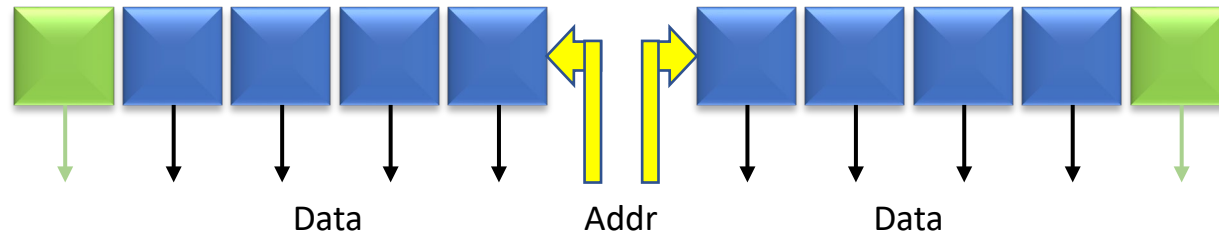


Flash Memory Summit



31.25 mm

133.35 mm



- Vertical sockets are typical for motherboards

JESD308 common specification

- 5V PMIC supply voltage
- Two 32- or 36-bit subchannels
- x8 or x16 SDRAMs (half of x8 ignored on ECC lane)
- Single data rate address/command bus
- One or two ranks supported with monolithics

JESD308	DDR5 Unbuffered Dual Inline Memory Module (UDIMM) Common Specification	
JESD308-U0-RCA	DDR5 Unbuffered Dual Inline Memory Module (UDIMM) Raw Card A Annex	1Rx8, 2 channel x 32b/ch
JESD308-U0-RCB	DDR5 Unbuffered Dual Inline Memory Module (UDIMM) Raw Card B Annex	2Rx8, 2 channel x 32b/ch
JESD308-U0-RCC	DDR5 Unbuffered Dual Inline Memory Module (UDIMM) Raw Card C Annex	1Rx16, 2 channel x 32b/ch
JESD308-U4-RCD	DDR5 Unbuffered Dual Inline Memory Module with 4-bit ECC (EC4 UDIMM) Raw Card D Annex	1Rx8 ECC, 2 channel x 36b/ch
JESD308-U4-RCE	DDR5 Unbuffered Dual Inline Memory Module with 4-bit ECC (EC4 UDIMM) Raw Card E Annex	2Rx8 ECC, 2 channel x 36b/ch

RDIMM: Registered Dual In-Line Memory Module

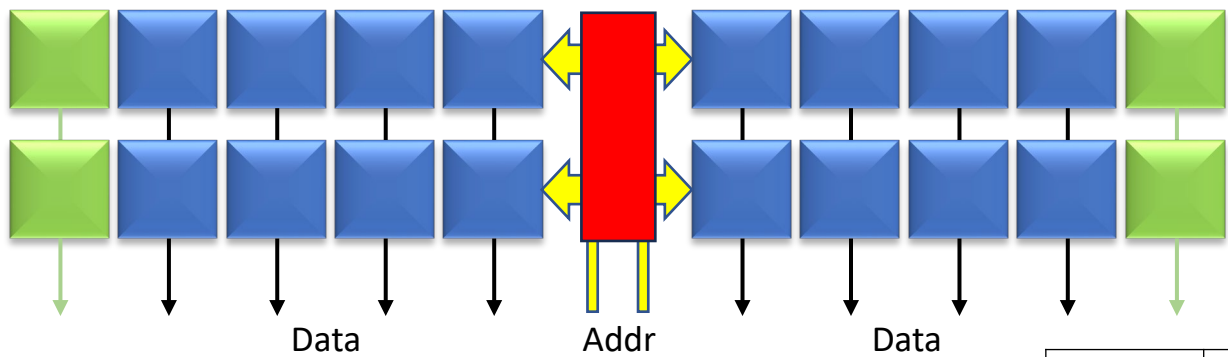
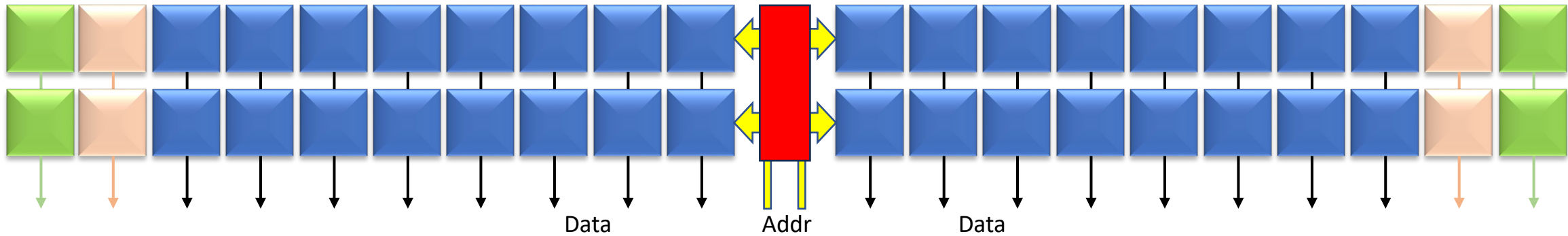


31.25 mm



Flash Memory Summit

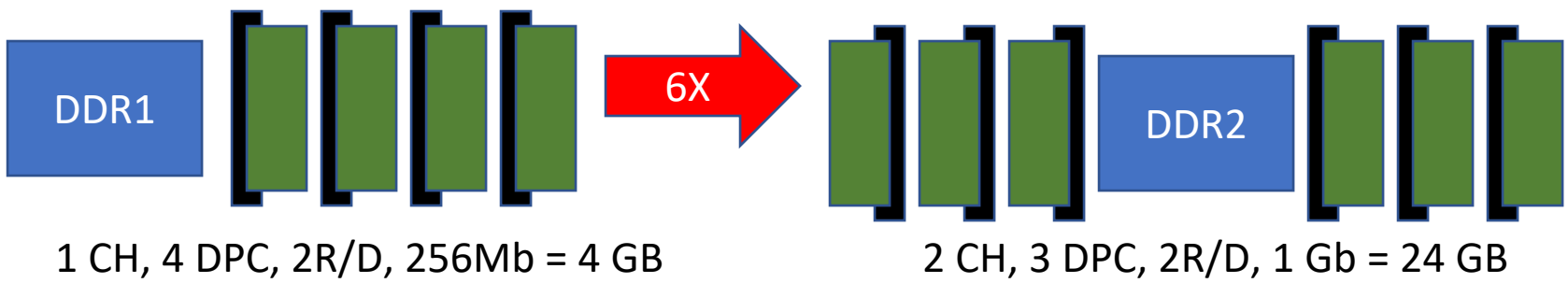
133.35 mm



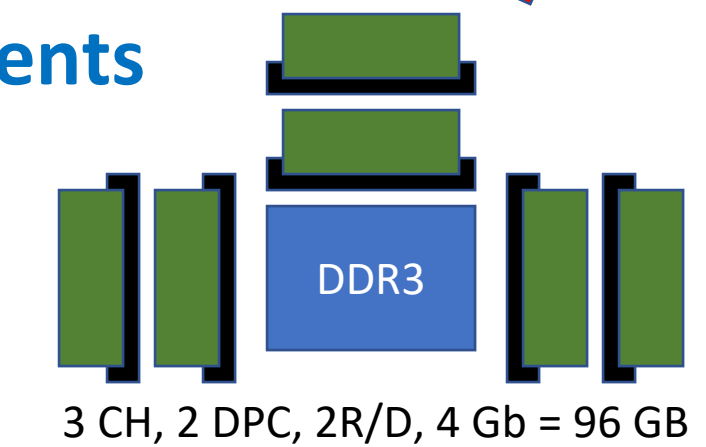
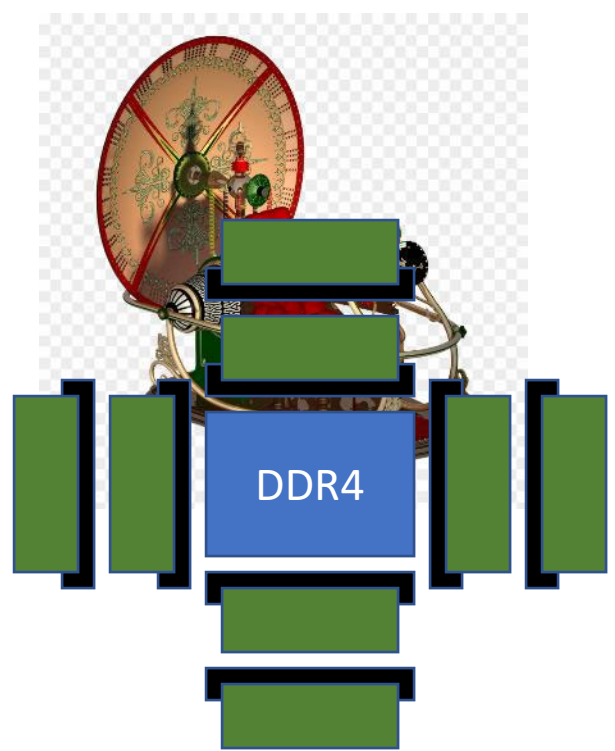
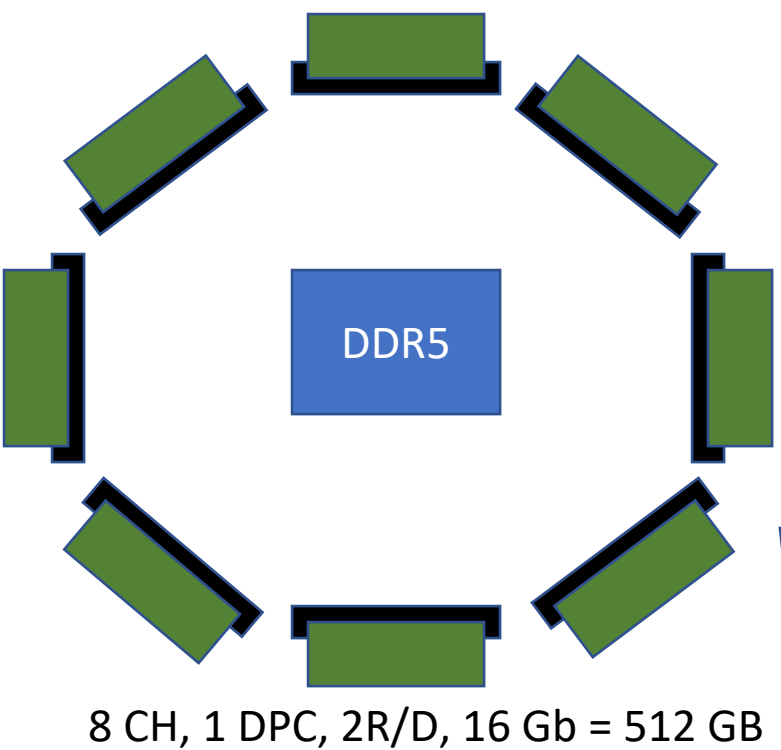
JESD305 common specification

- 12V PMIC supply voltage
- Two 36- or 40-bit subchannels
- x4 or x8 SDRAMs
- Double data rate skinny address/command bus
- One or two ranks supported

JESD305	DDR5 Load Reduced (LRDIMM) and Registered Dual Inline Memory Module (RDIMM) Common Specification	
JESD305-R8-RCA	DDR5 Registered Dual Inline Memory Module with 8-bit ECC (EC8 RDIMM) Raw Card A Annex	2/4/8/16Rx4, 2 channel x 40b/ch
JESD305-R4-RCB	DDR5 Registered Dual Inline Memory Module with 4-bit ECC (EC4 RDIMM) Raw Card B Annex	2/4/8/16Rx4, 2 channel x 36b/ch
JESD305-R8-RCC	DDR5 Registered Dual Inline Memory Module with 8-bit ECC (EC8 RDIMM) Raw Card C Annex	1/2/4/8Rx4, 2 channel x 40b/ch
JESD305-R8-RCD	DDR5 Registered Dual Inline Memory Module with 8-bit ECC (EC8 RDIMM) Raw Card D Annex	1Rx8, 2 channel x 40b/ch
JESD305-R8-RCE	DDR5 Registered Dual Inline Memory Module with 8-bit ECC (EC8 RDIMM) Raw Card E Annex	2Rx8, 2 channel x 40b/ch
JESD305-R4-RCF	DDR5 Registered Dual Inline Memory Module with 4-bit ECC (EC4 RDIMM) Raw Card F Annex	1Rx4, 2 channel x 36b/ch

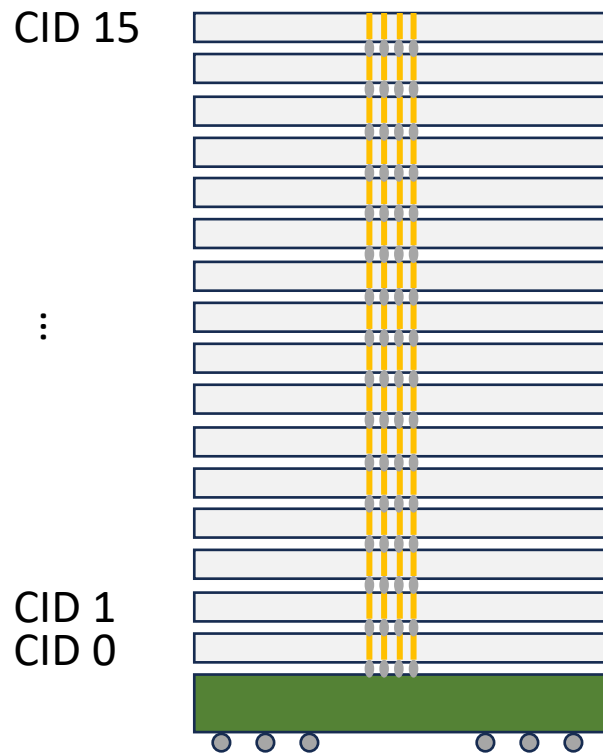


Increasing frequency is slowing DIMM improvements



CH = channel
DPC = DIMMs per channel
R/D = ranks per DIMM
Assumes no 3DS

3DS to the Rescue!



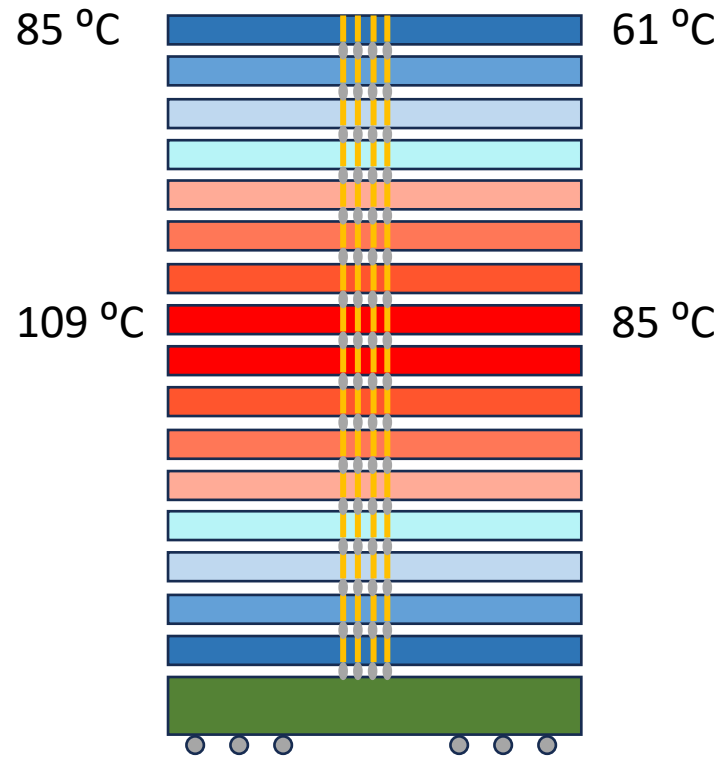
We can stack 16 DRAMs in one package!

- 16 x 16 Gb dies → 32 GIGABYTES per stack! Hooray!
- That 64 GB DIMM suddenly becomes a 1 TB DIMM!
- Thin the die to expose through-silicon vias (TSVs)!
- Microbump or high density interconnect them!
- The bottom die will proxy the stack for only 1 load!



Almost too good to be true!

Ummmm...



Flash Memory Summit

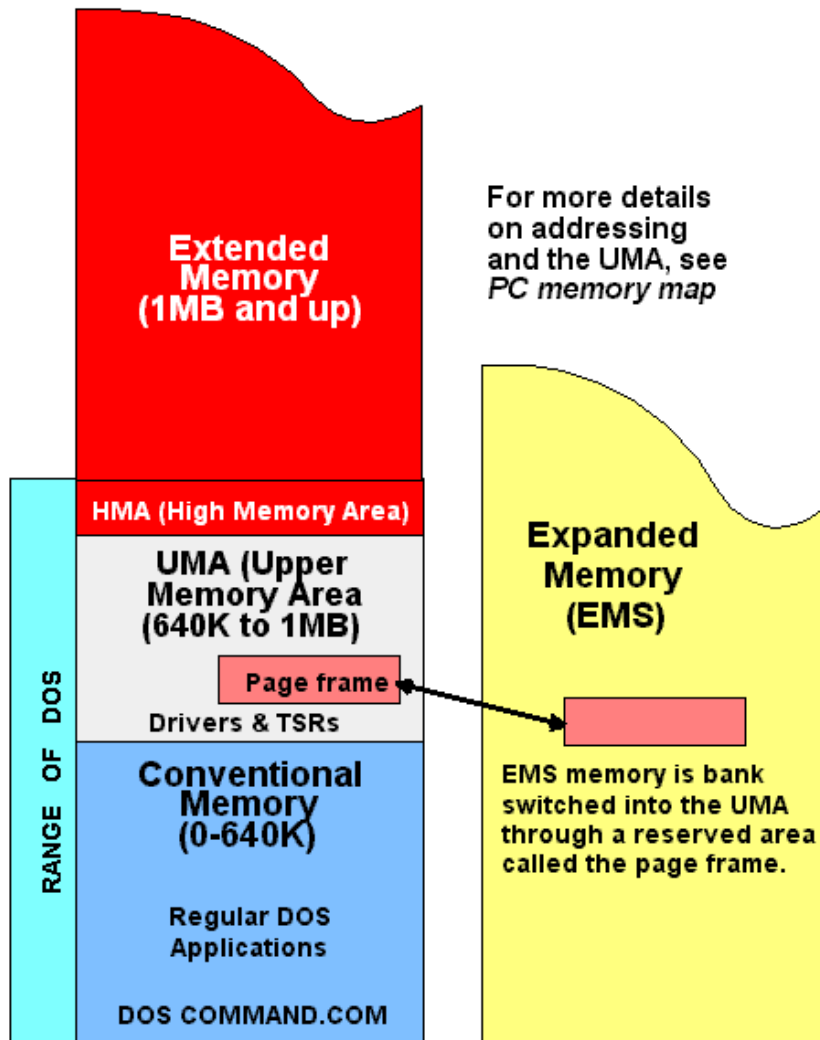
Reality check on 3DS

- The middle dies get really, **really** hot
- Compound die yield is a problem
 - 97% yield per die is nice
 - $0.97^{16} = 61\%$ yield
 - And this assumes all die speed bin equivalently fast
- Manufacturability of 3DS continues to drag
- Refreshing 16 die? Really???
- 3DS dropped to 8 die, then to 4 die, then...

3DS is giving way to dual die package (DDP)

- Simpler assembly using more standard methods such as redistribution layers
- Die thinning is optional
- Requires new logic support





Memory Expansion is Not New

In the 1980s, Expanded and Extended Memory were common methods to grow the memory footprint of a PC beyond the CPU limits

Real time operating systems running on such systems had to comprehend the differences in access times

Memory Pooling is Also Not New

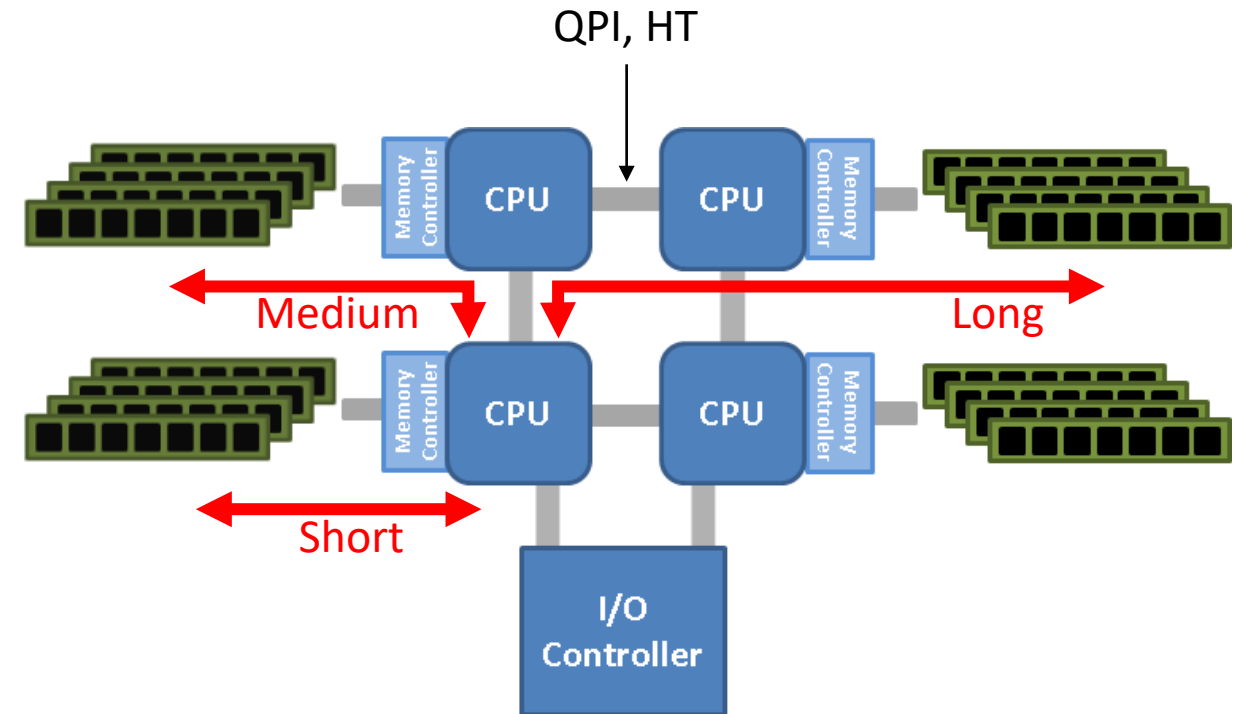
Non-Uniform Memory Architectures (NUMA) have been common ways to pool memory resources

Buses such as HyperTransport and Quick Path Interconnect have been around for decades

These NUMAs created a tier of resources

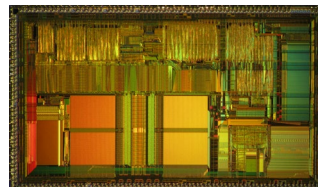
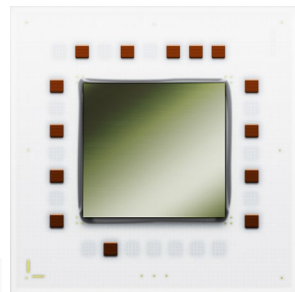
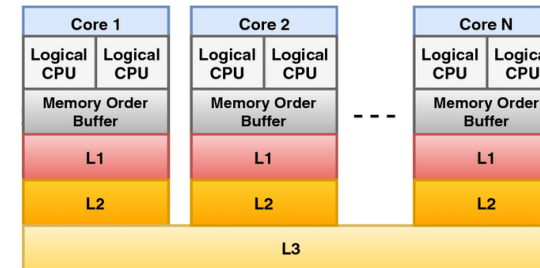
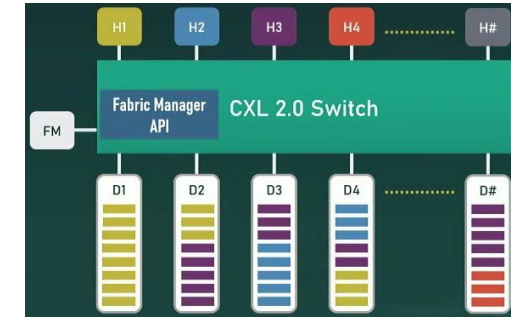
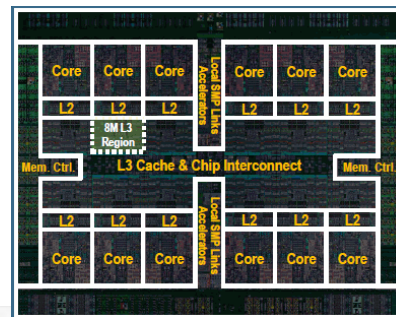
- Fastest memory attached to CPU
- Slower memory one hop away
- Slowest memory two hops away

Smart software adjusted data location based on access latency





As CPUs grew hungrier

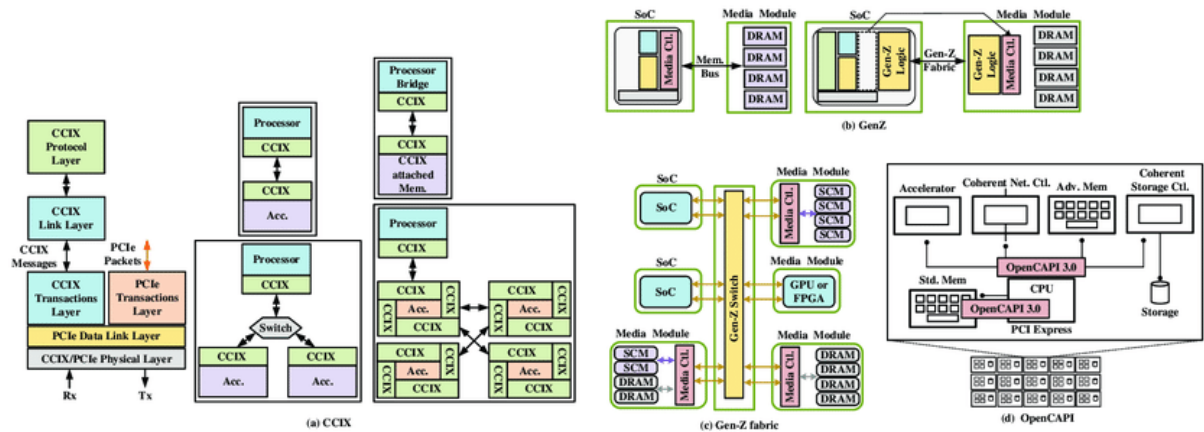


Memory solutions grew deeper and more complex



Fabric Wars

Proprietary fabrics emerged for resource sharing, however lack of standardization limited the audience

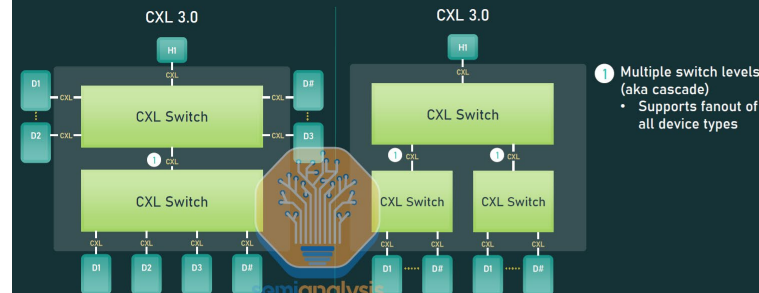


CXL Big Bang

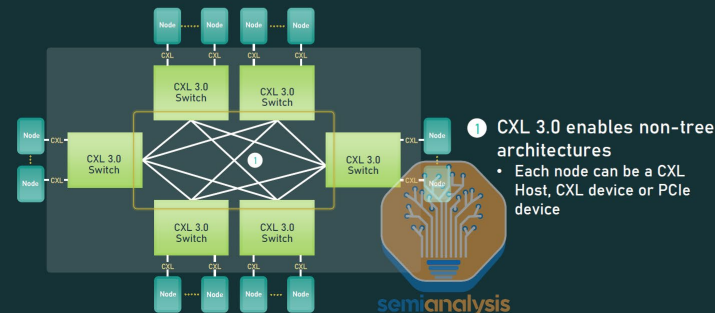
Wide adoption of CXL allows for standardization and commoditization of expansion resources and sharing

CXL 3.0: SWITCH CASCADE/FANOUT

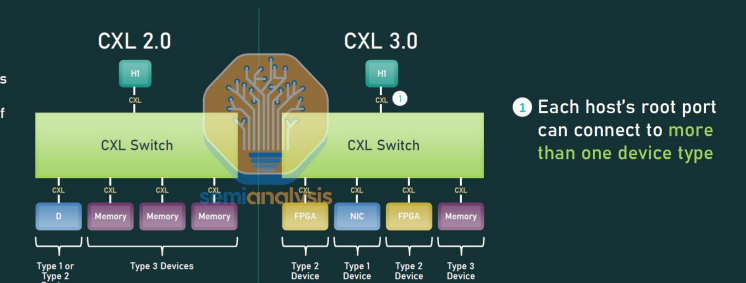
Supporting vast array of switch topologies



CXL 3.0: FABRICS OVERVIEW

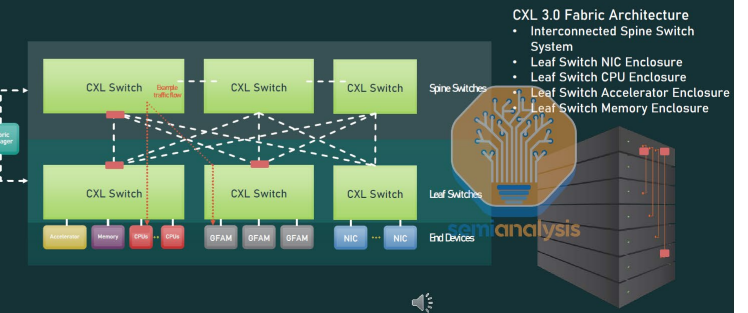


CXL 3.0: MULTIPLE DEVICES OF ALL TYPES PER ROOT PORT



CXL 3.0: FABRICS EXAMPLE USE CASE

Composable Systems with Spine/Leaf Architecture



Why Put DRAM on CXL?



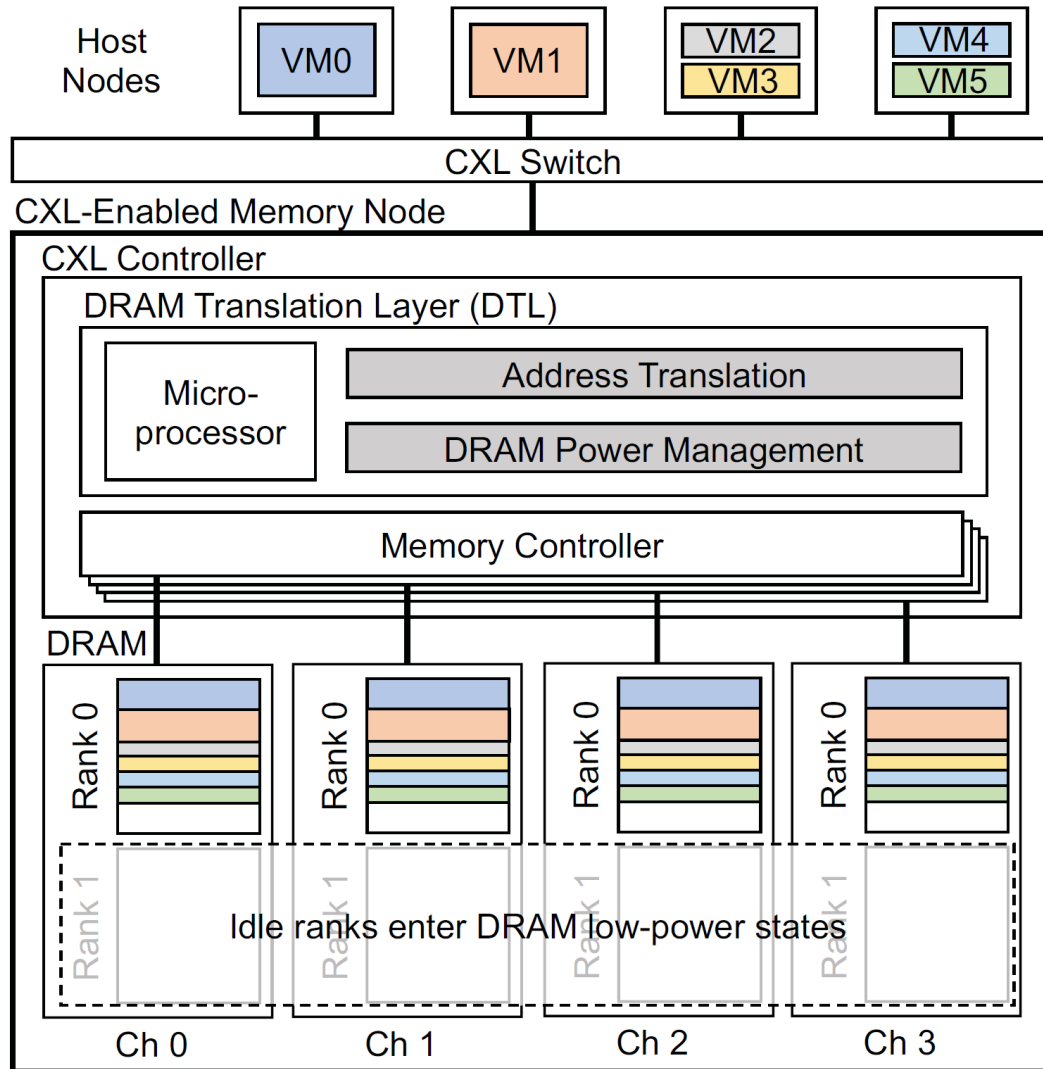
Flash Memory Summit

DDR5 → 1 DIMM/channel
DRAM stalls at 32Gb
AI demands more memory
Sales team whines about
having nothing to sell



CXL enables nearly unlimited
memory expansion
Memory pooling allows
unused memory to be
reallocated

Not to be rude, but
what choice do you
really have?



<https://dl.acm.org/doi/abs/10.1145/3579371.3589051>

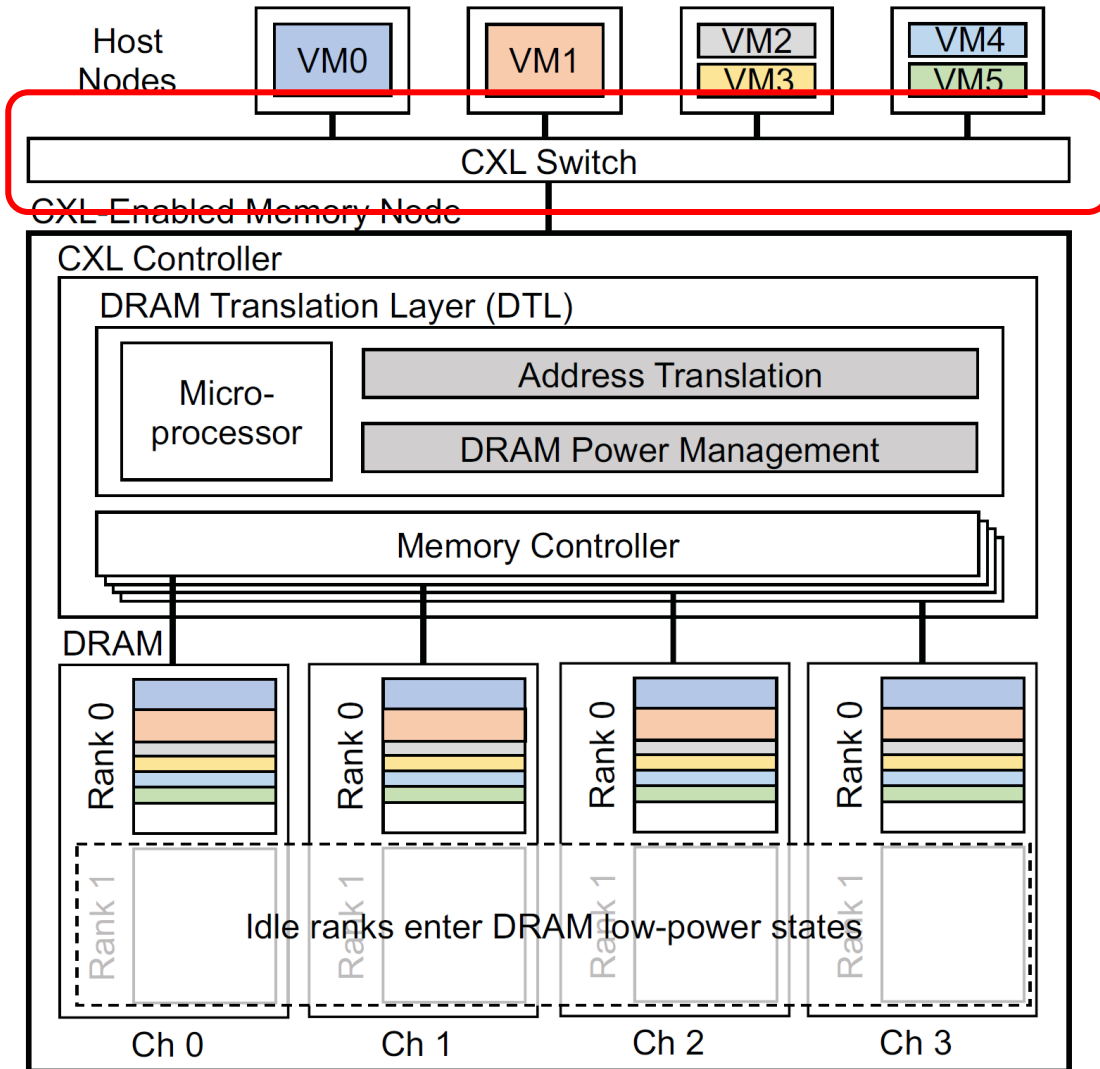
CXL Unifies the Fabric

CXL is PCIe based and therefore inherits some of the features and limitations of a protocol that supports I/O or memory expansion

Legacy software only had filesystems to implement virtualization – DAX is assisting movement towards a unified addressing structure, but...

...is DAX stalled with the death of Optane?

...will CXL semantics breathe new life into a unified memory model?



<https://dl.acm.org/doi/abs/10.1145/3579371.3589051>

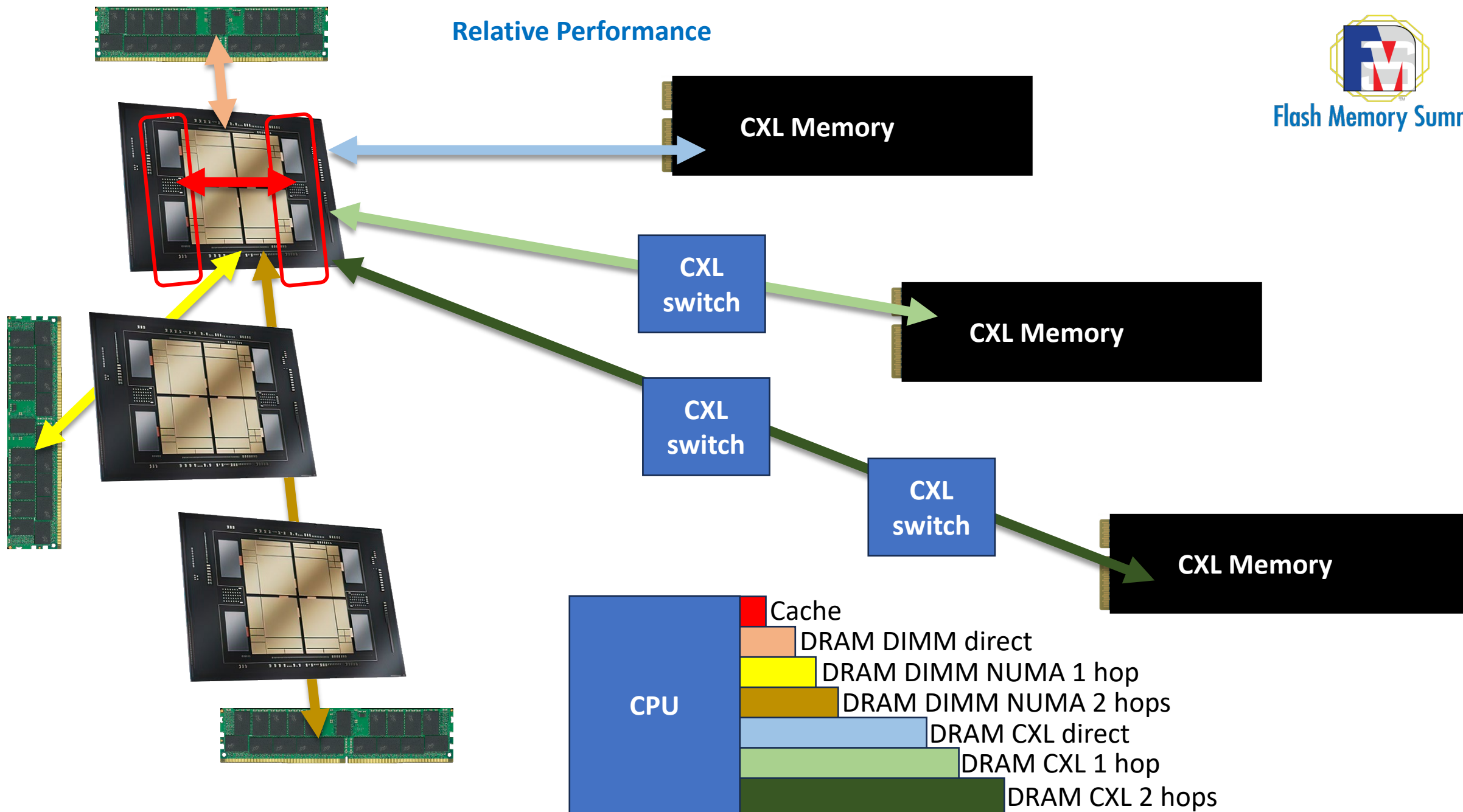
CXL Switches

CXL switches are likely going to be the next “fabric war” as it fragments into dumb hubs versus highly intelligent controllers

A big hole in CXL 3.0 is the lack of definition of a “fabric manager”

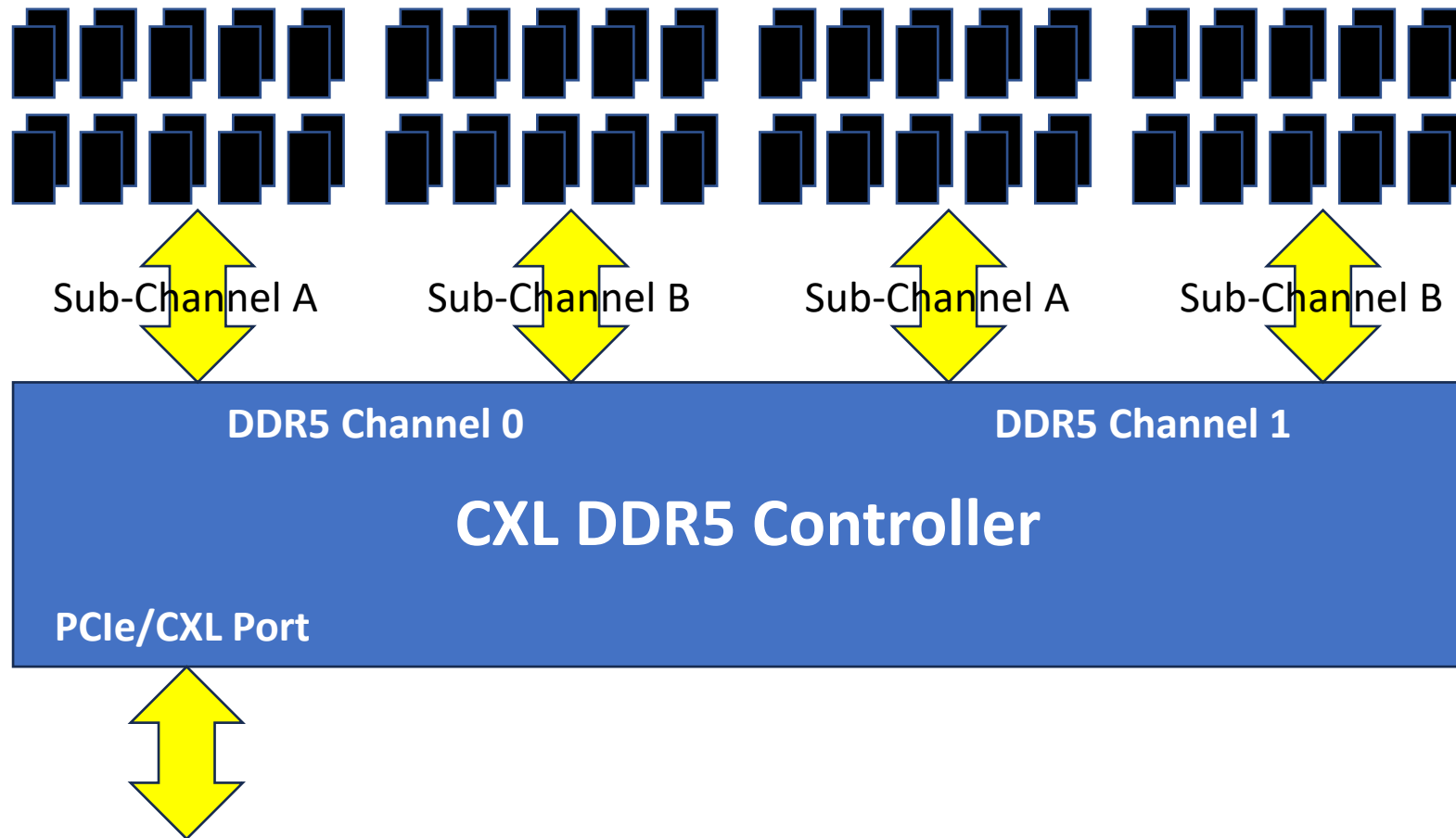
For now, except in a few places, we can ignore the switch

Relative Performance





Anatomy of a CXL to DRAM Bridge

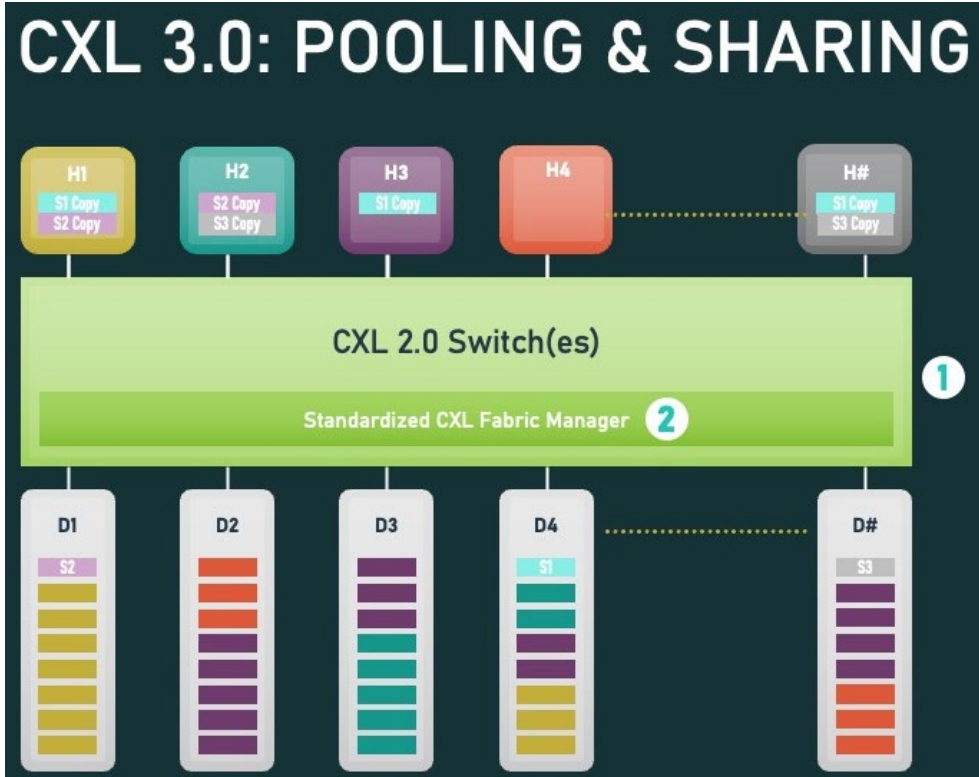


KISS: Just Do Writes and Reads

CXL is a non-deterministic protocol which allows the CXL module to operate independently

- Refresh
- Error check scrub
- Post-package repair

CXL 3+ incorporates some additional functions such as coherency



It's a Brave New World with CXL Memory

CXL memory modules may be dedicated to a single processor

CXL memory modules may be allocated in chunks to different processors

CXL memory modules may be shared by multiple processors

But What About Cache Coherency Via Back Invalidation?

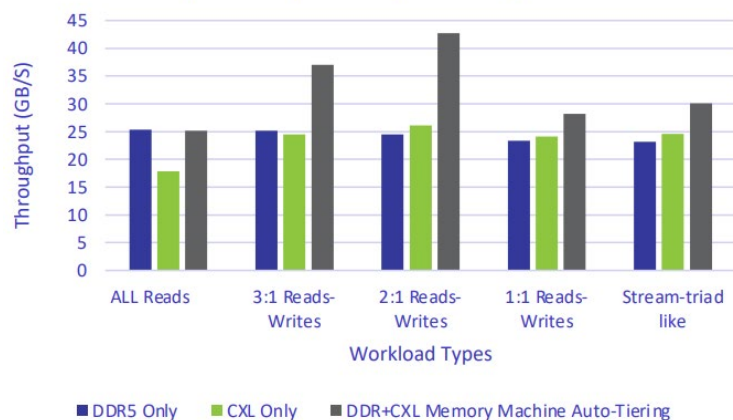
Someone smarter than me needs to explain how back invalidation works if a CXL memory region is shared by a random number of CPUs...

Drivers, e.g., Memory Latency Checker

Operating systems measure the access latency of the various memory regions, categorize them

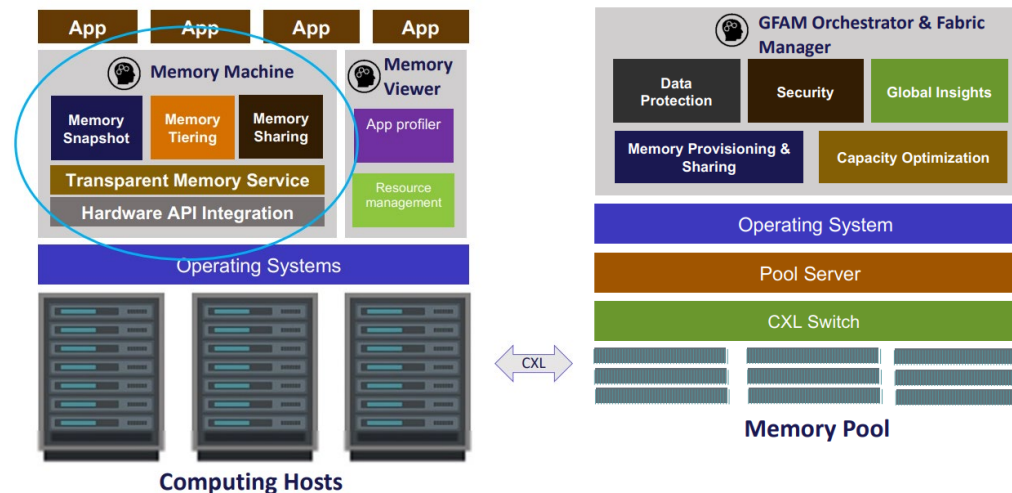
```
Measuring idle latencies (in ns)...
Memory node
Socket 0 1
0 67.5 125.2
1 126.5 68.5
```

MLC (Memory Latency Checker) Results



Hypervisors, e.g., MemVerge

Runtime monitoring of system resource utilization and characterization of hot/warm/cold data



Operating System Support

Linux kernel support memory hotplug & hotremove today

Need Dynamic Capacity Driver in Linux kernel

Policy should be implemented in userspace

- When to request memory (hotplug)
- When to release memory (hotremove)

OS improvement to make hotplug & hotremove faster (keep region map, ...)

OS improvement to avoid memory pinning (which block hotremove)

- Linux kernel already have some of that (zone movable comes with tradeoff)



Let's talk about POWER



U.S. DEPARTMENT OF
ENERGY

Office of
ENERGY EFFICIENCY &
RENEWABLE ENERGY

ADVANCED MATERIALS &
MANUFACTURING
TECHNOLOGIES OFFICE

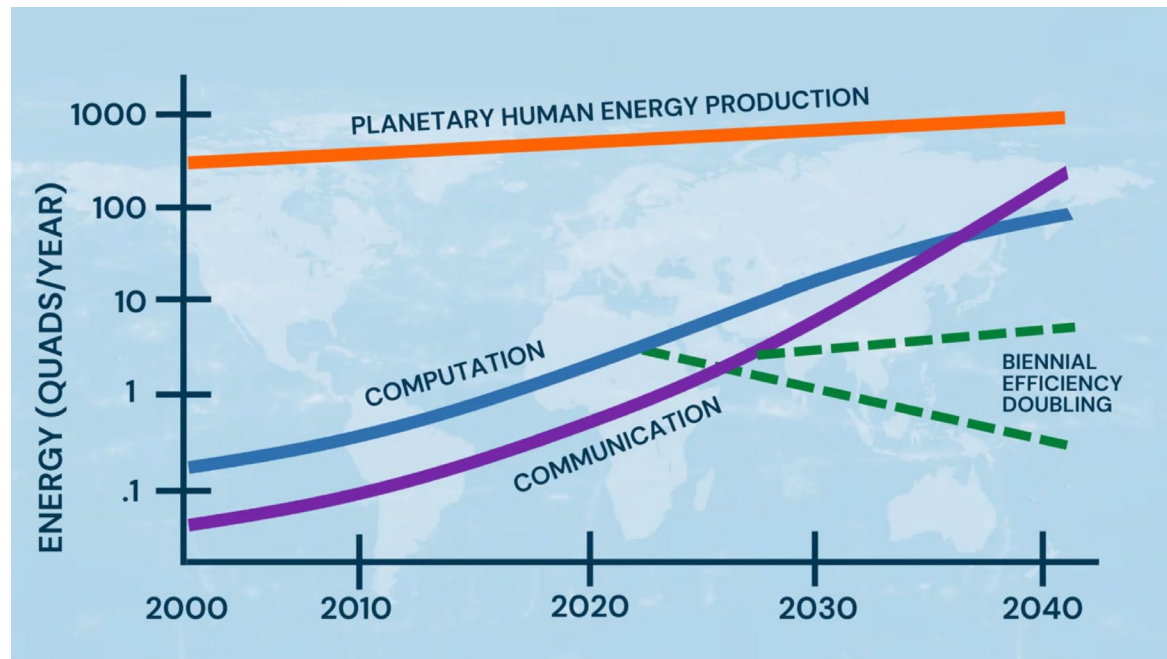


Flash Memory Summit

Designing for energy efficiency is a growing concern

“Total Cost of Ownership” partially encompasses this





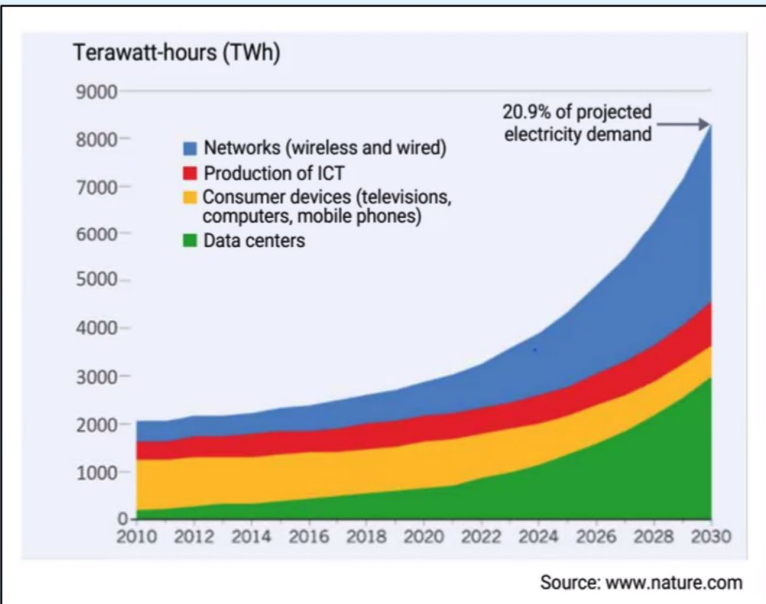
On the current trajectory of energy use versus energy production,

THESE CROSS OVER IN 2055

EES2 program goal is 1000X improvement in energy efficiency over the next 20 years

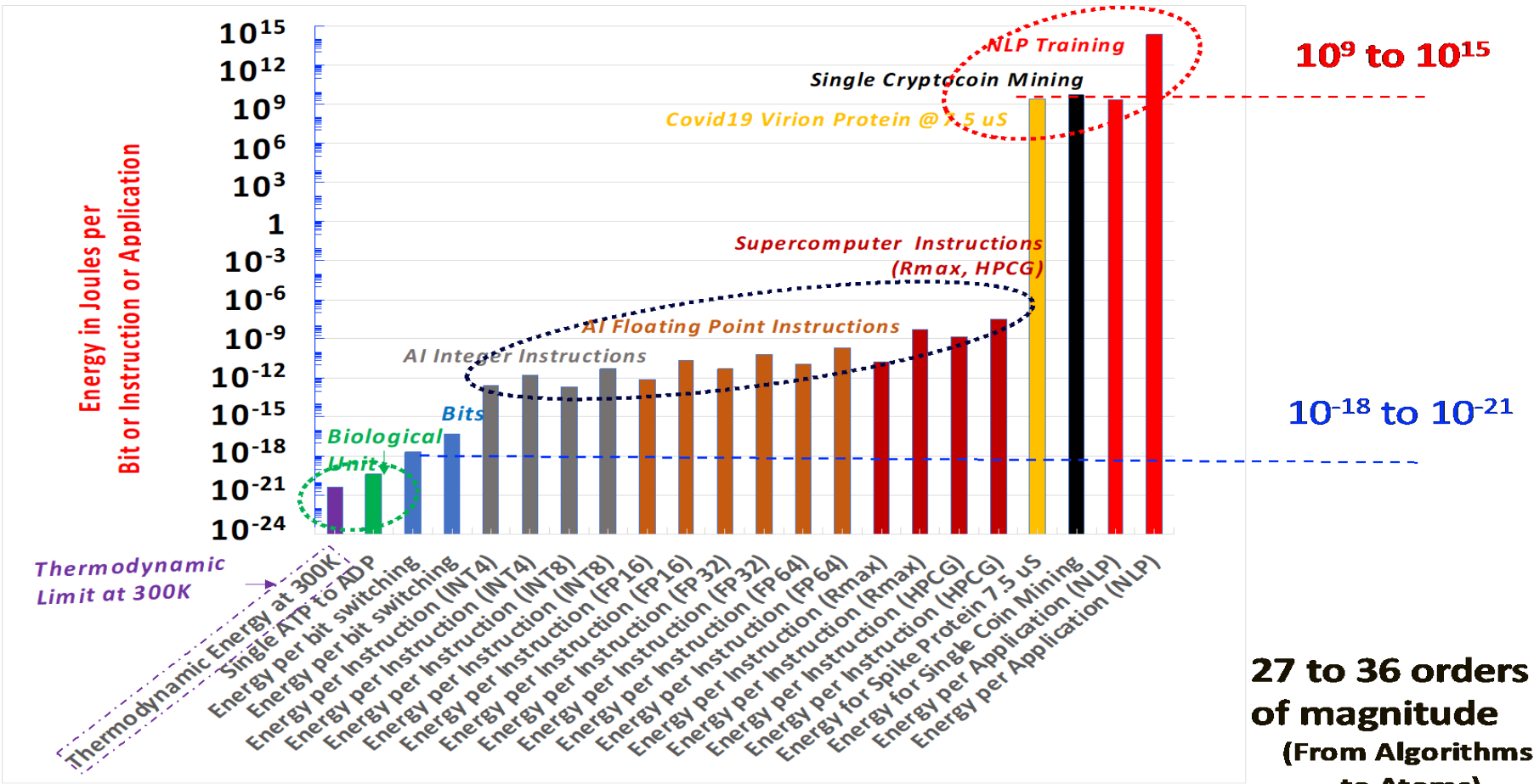
This program is not US-centric
All countries are invited to participate

This program is tied into the US
CHIPS Act funding



Operation	Energy per bit
Wireless data	10 – 30μJ
Internet: access	40 – 80nJ
Internet: routing	20nJ
Internet: optical WDM links	3nJ
Reading DRAM	5pJ
Communicating off chip	1 – 20 pJ
Data link multiplexing and timing circuits	~ 2 pJ
Communicating across chip	600 fJ
Floating point operation	100fJ
Energy in DRAM cell	10fJ
Switching CMOS gate	~50aJ – 3fJ
1 electron at 1V, or 1 photon @1eV	0.16aJ (160zJ)

most energy is used for communications, not logic



Part of the looming energy crisis is fundamental inefficiencies of applications and programming languages

Python programming is orders of magnitude less energy efficient than C programming (ChatGPT is Python-based)

Cryptocurrency in particular consumes $\geq 0.5\%$ of world energy resources already

27 to 36 orders of magnitude
(From Algorithms to Atoms)



Education is needed



System OEMs need to agree this crisis is costing them

Total cost of ownership analyses need to account for power costs

Suppliers need to agree there are solutions to this problem

Standards bodies need to participate

Government support for these changes is essential

Universities need to engage to get the next generation involved



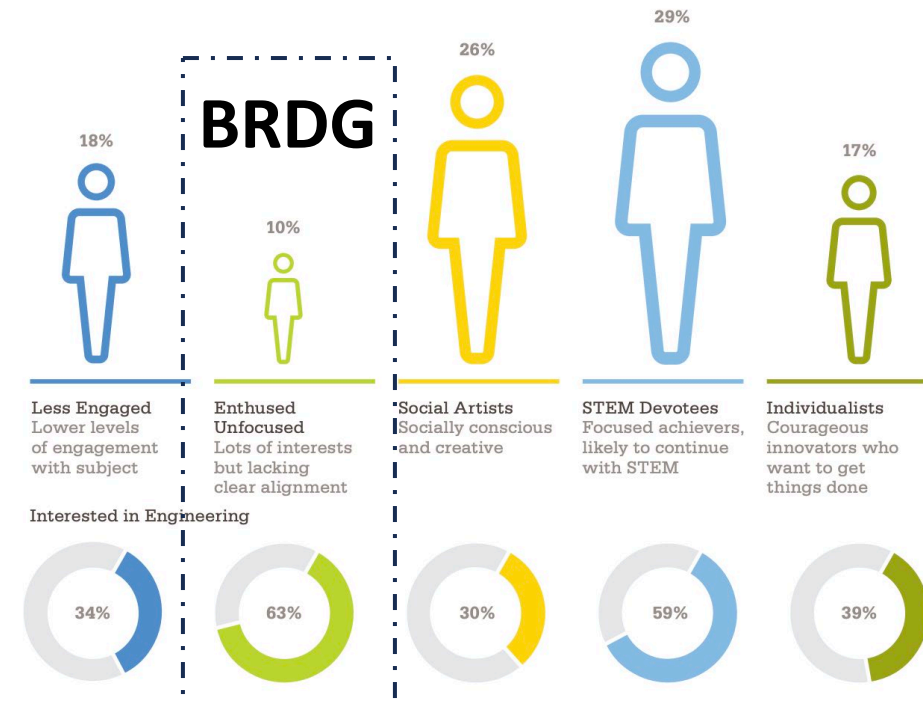
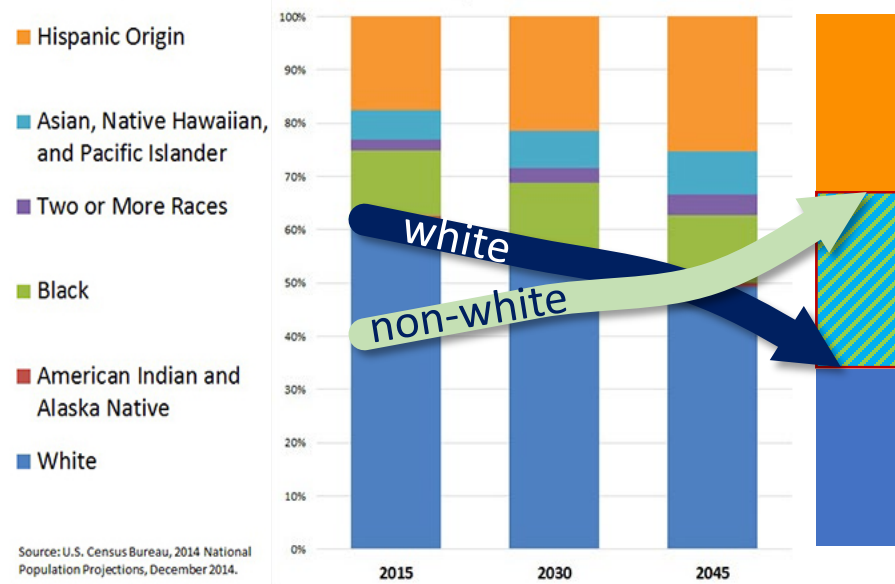
Flash Memory Summit

Recognize the changing US demographics

Goal is to enhance career skills & opportunities for students from first-in-family @ college backgrounds

Connect students to EES2 program (projects, posters, etc)

Education non-profit
Mentoring university students in STEM
Member of EES2



<https://www.imeche.org/policy-and-press/reports/detail/five-tribes-personalising-engineering-education>



Flash Memory Summit

Let's make a difference

- Mentor
- Coach
- Support



bridge-to-connect.org



BRDG develops students into future leaders

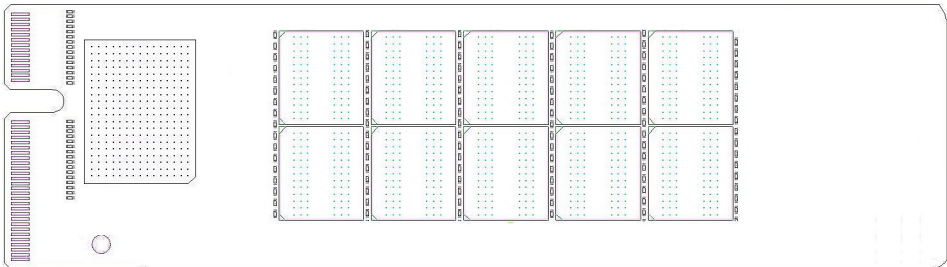


RDIMM



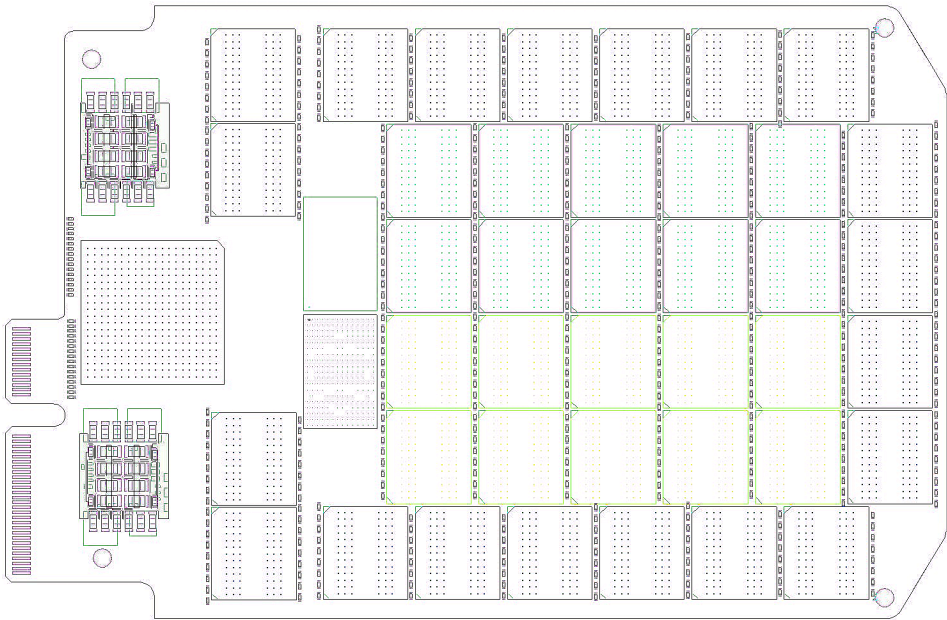
32GB
20 DRAMs
64GB
40 DRAMs

**CXL
E1.S**



16GB
10 DRAMs
32GB
20 DRAMs

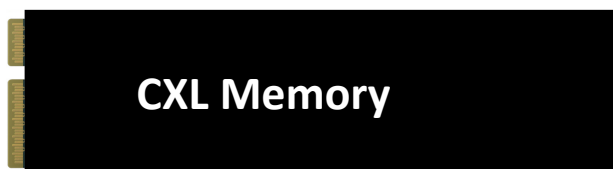
**CXL
E3.S**



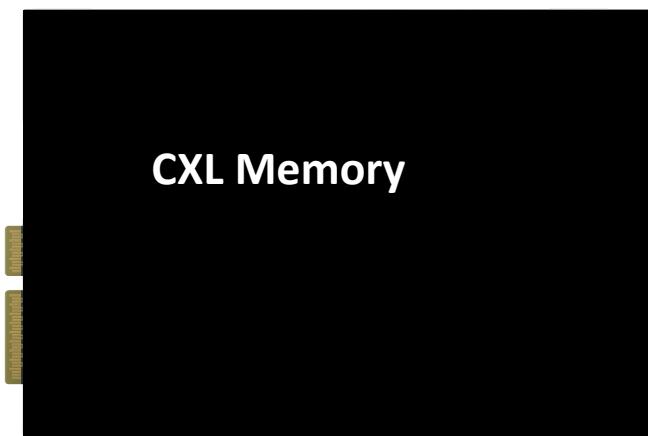
64GB
40 DRAMs
128GB
80 DRAMs



DDR5-6400 RDIMM



DDR5 CXL Memory Module, E1.S



DDR5 CXL Memory Module, E3.S

32GB

Power = 7.5W
RCD, PMIC @ 3W
20 DRAMs @ 4.5W

64GB

Power = 12W
RCD, PMIC @ 3W
40 DRAMs @ 9W

16GB

Power = 10.25W
CMC, PMIC @ 8W
20 DRAMs @ 2.25W

32GB

Power = 12.5W
CMC, PMIC @ 8W
20 DRAMs @ 4.5W

64GB

Power = 17W
CMC, PMIC @ 8W
40 DRAMs @ 9W

128GB

Power = 26W
CMC, PMIC @ 8W
80 DRAMs @ 18W

Power Util.

4.3 GB/W

5.3 GB/W

1.6 GB/W

2.6 GB/W

3.8 GB/W

4.9 GB/W

Goodness

81%

100%

29%

49%

72%

92%

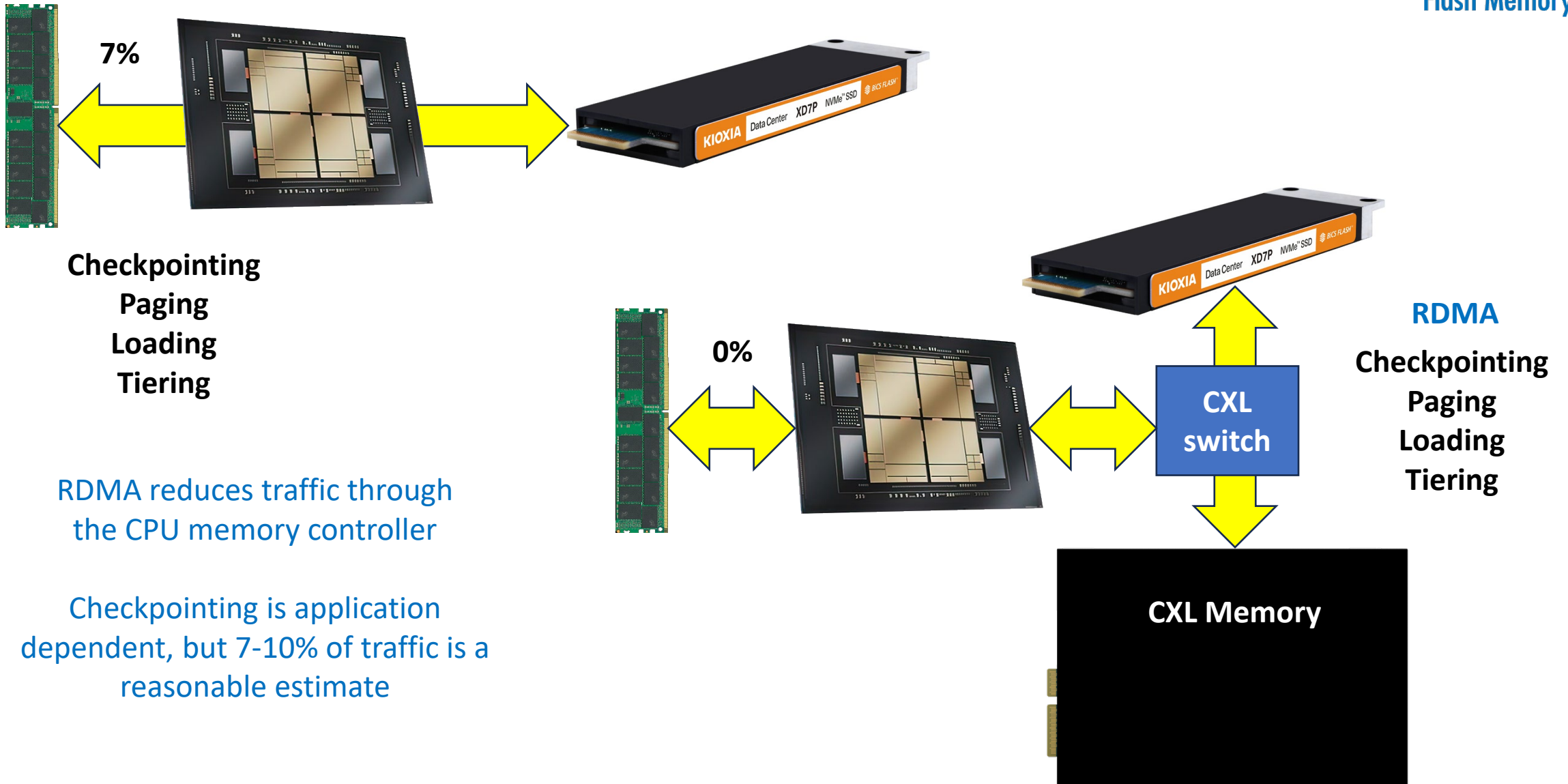
Metric used:

How many gigabytes do you get
for every watt you expend?

Conclusions:

E1.S form factor is hard to justify
E3.S w/80 DRAMs close to DIMM
power

Performance Enhancement from Remote Direct Memory Access (RDMA)



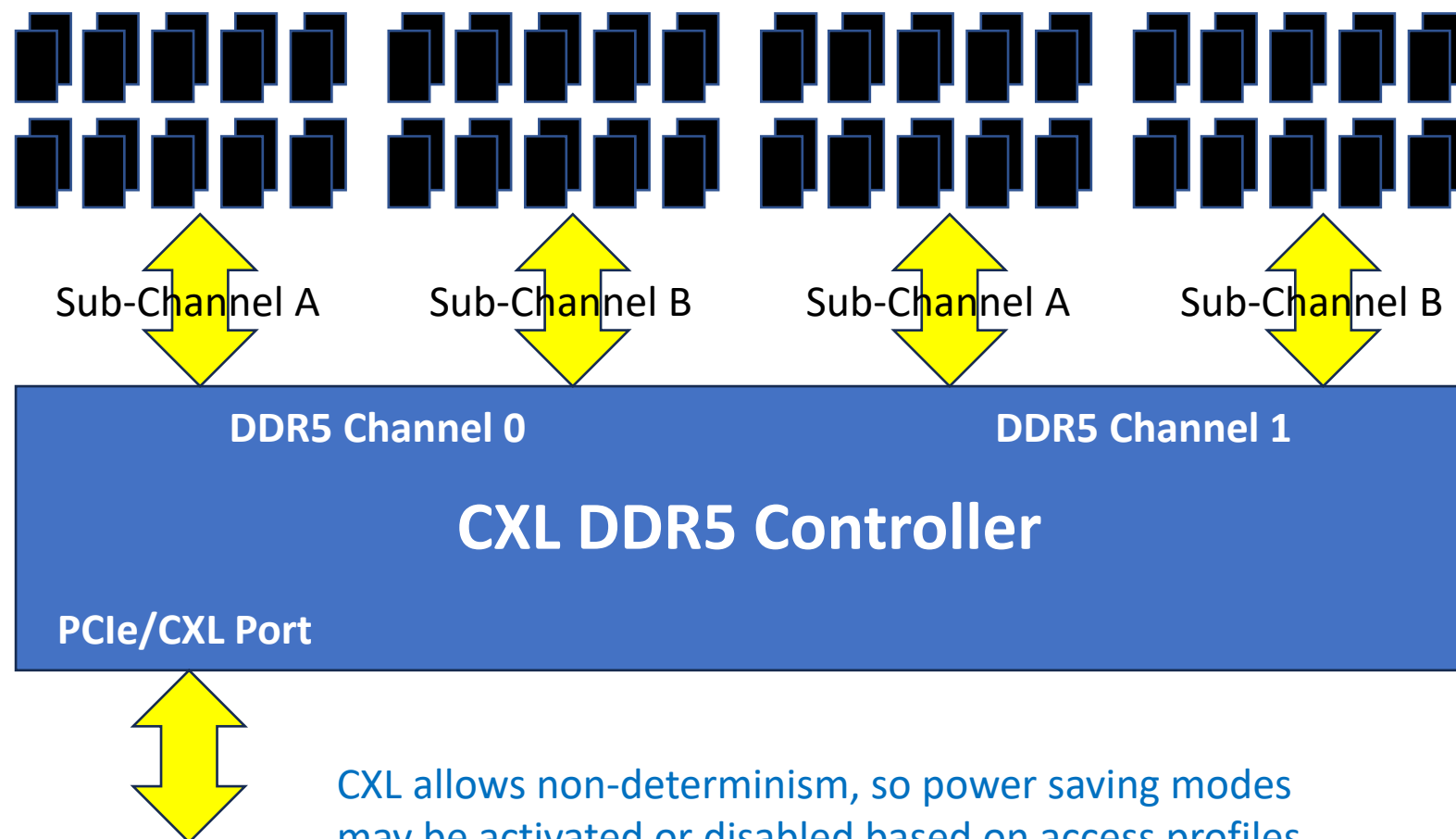


Power	Definition	DDR4 mA	Norm
IDDO	Active precharge	31	1.9
IDD1	Active read precharge	44	2.8
IDD2P	Precharge power-down	16	1.0
IDD3P	Active power-down	21	1.3
IDD2N	Precharge standby	22	1.4
IDD3N	Active standby	36	2.3
IDD4R	Read current	101	6.3
IDD4W	Write current	84	5.3
IDD5	Refresh	199	12.4
IDD6	Self-refresh	23	1.4
IDD7	Bank interleave read	142	8.9

Where are we spending our power?

Some simplified looks:

Refresh burns >10X idle power
Activate uses 11%
Precharge uses 21%



CXL allows non-determinism, so power saving modes may be activated or disabled based on access profiles, user configuration settings, etc.

Mode switching latency penalty need only be taken once – what's a microsecond when a region has not been accessed for an hour?

Optimizing DRAM power

Use closed page mode to avoid active standby power penalty

Use CKE & self-refresh for memory regions not used often

Use Maximum Power Saving Mode for DRAM not yet allocated



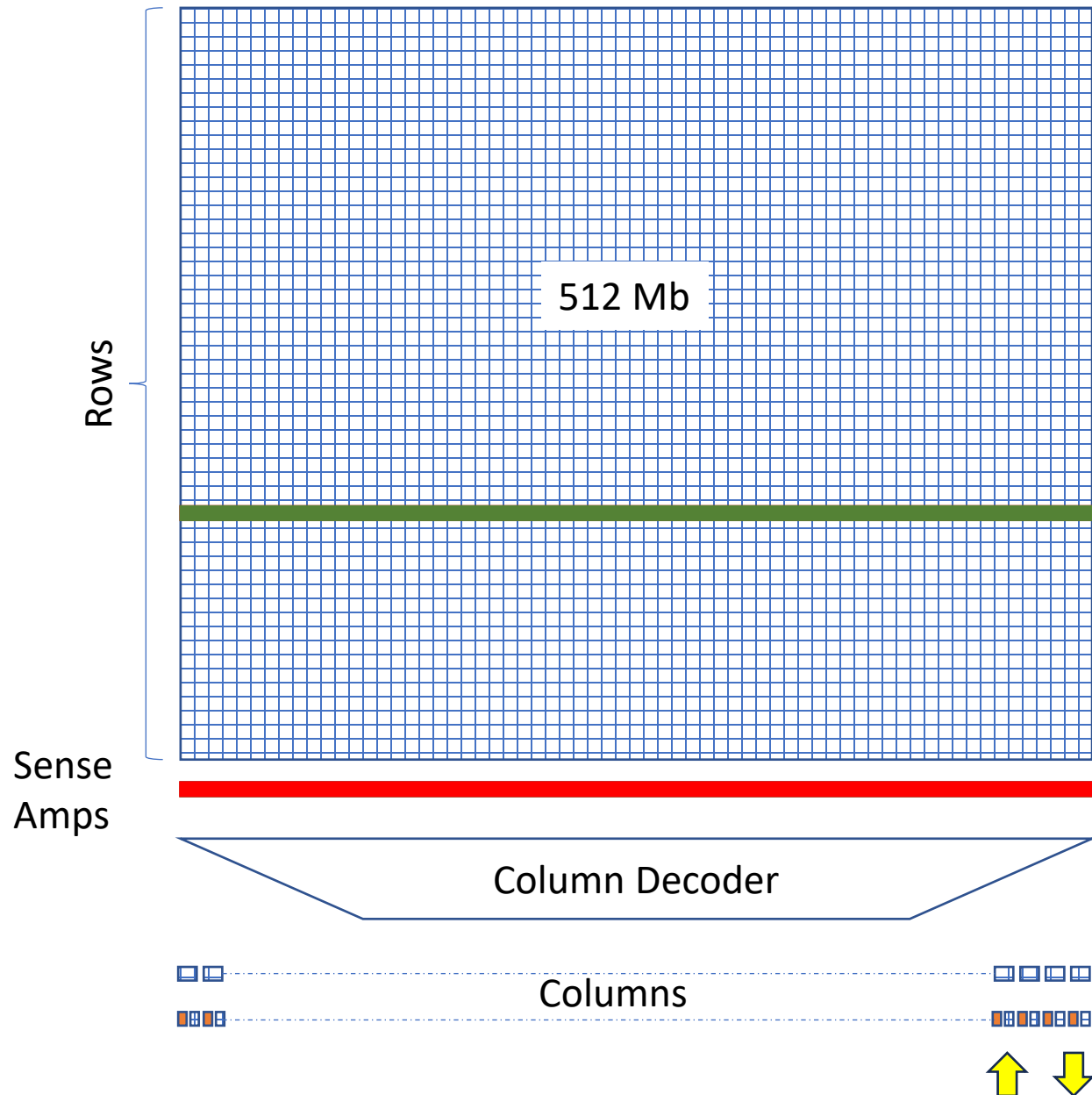
DRAM access procedure:

ACTIVATE reads 8192 from core to sense amps, destroying the contents of the core bits

READ operations transfer 128 bits (x8) or 64 bits (x4) from sense amps to the I/O

Write operations transfer 128 or 64 bits from I/O to sense amps

PRECHARGE rewrites 8192 bits back to the core



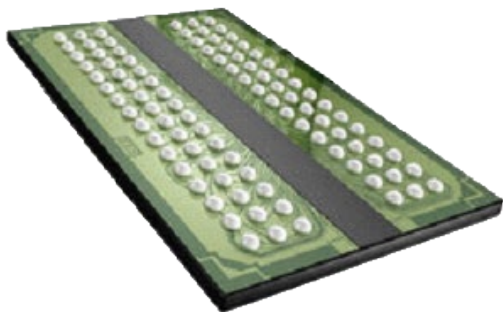
	%Of array
8192 b	1.5%

	%Of buffer	%Of array
128 b	1.5%	0.025%
64 b	0.8%	0.012%

Conclusion: Closed Page Mode access is grossly inefficient

All bits used for x8 DRAMs
ECC half-word needed for x4

CXL Opens the Door to CXL-Optimized Memory Designs



Standard DDR5

Open & closed page modes

1 KB page size per bank

Complex DFE for many system configurations

Complex ODT for many system configurations

64GB

Power = 17W

CMC, PMIC @ 8W

40 DRAMs @ 9W

Power Util.

3.8 GB/W

Goodness

72%

128GB

Power = 26W

CMC, PMIC @ 8W

80 DRAMs @ 18W

4.9 GB/W

92%



CXL-Optimized DDR5

Closed page mode only

Arbitrary page size, as little as 128 bits

Simplified DFE for restricted system configurations

Simplified ODT for restricted system configurations

64GB

Power = 11W

CMC, PMIC @ 8W

40 DRAMs @ 6W

Power Util.

5.8 GB/W

Goodness

109%

128GB

Power = 20W

CMC, PMIC @ 8W

80 DRAMs @ 12W

6.4 GB/W

121%



43% of data center power is used for servers →

20% of data center power is used for DRAM

43% of power is used for cooling

**Use of CXL optimized DRAM could save 30%
of DRAM power = 6% of data center power
+ 2% savings on cooling**



Summary

DRAM evolution is slowing down

Most changes in DDR1-DDR5 address signal integrity

Refresh is harsh penalty and security forces more

DDR5 is highly configurable – hundreds of mode registers

DDR5 improves transparency regarding errors and repair

DRAMs and memory modules are co-designed

DDR5 modules incorporate SidebandBus and PMICs

Serial Presence Detect chip becomes a hub, helps set up

The world is reaching a crisis point in energy use

CXL allows for memory expansion

CXL opens the door to power efficient memory

DRAM design is an evolution of 1990s SDRAM

Thank you for your time

Any more questions?

Bill Gervasi, Principal Systems Architect
Wolley Inc.
bilge@wolleytech.com

