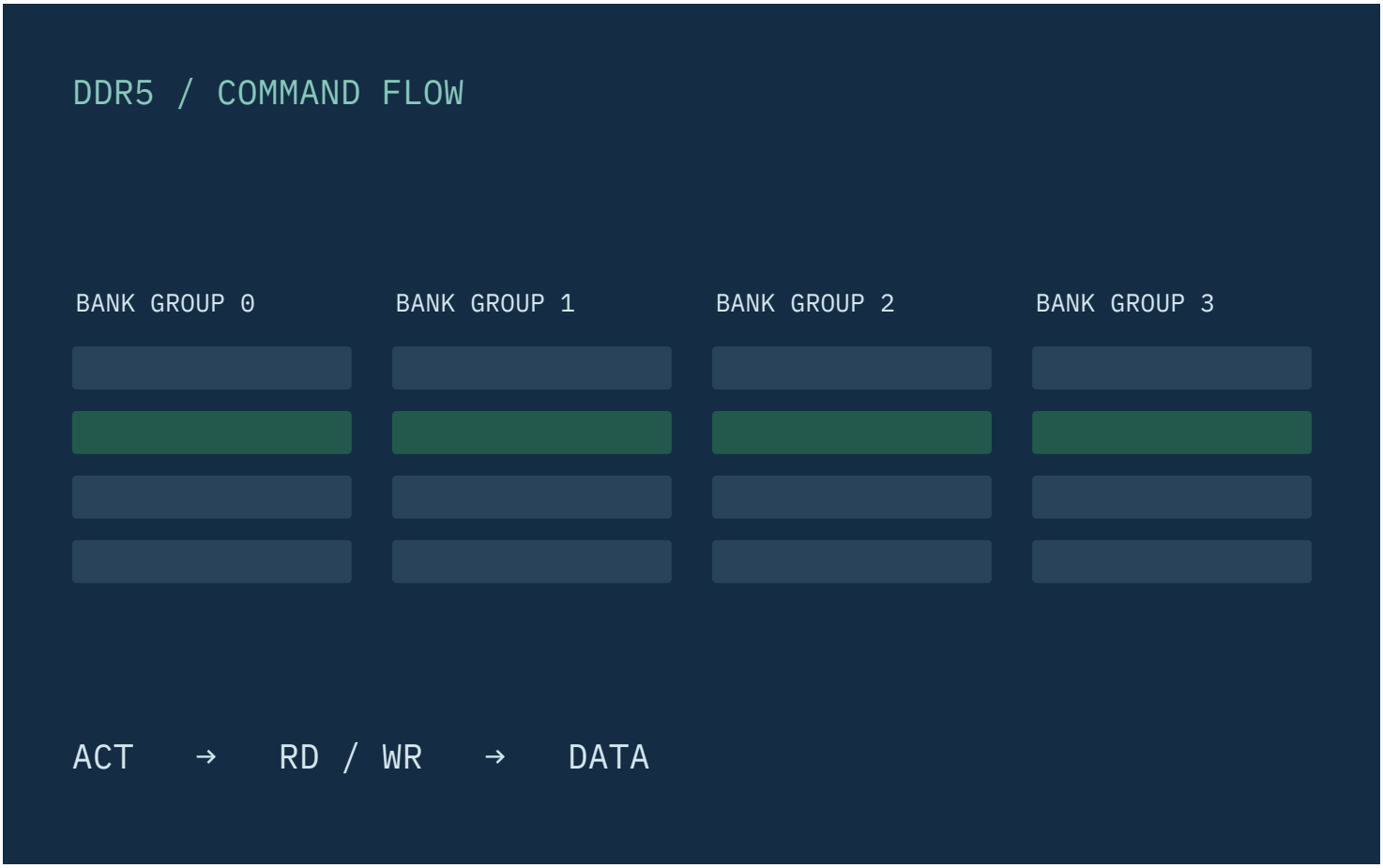
2026



DDR5: PRINCIPLES OF MEMORY OPERATION

Architecture, Commands, and Timings

CELLS / ROWS / COMMANDS / DATA REFRESH

A standalone theory volume in the DDR5 manual series

VERSION 1.01 (05.11.2026)

English edition · From memory architecture to commands and timings

M. D. PAKHTUNOV / T.ME/PXTNV | T.ME/PXTECH

© Maksim Pakhtunov, 2026 · All rights reserved

Contents

01. Introduction to DDR5 theory . . . 5
1.1. Storage locations for a running program's data . . . 5
1.2. Bits, bytes, and basic memory characteristics . . . 5
1.3. Main topics of the guide . . . 6
1.4. Guide scope and reading sequence . . . 7
02. The DRAM cell, sensing, and charge restoration . . . 8
2.1. Charge, a switch, and array lines . . . 8
2.2. Row activation . . . 8
2.3. The open row and row buffer . . . 9
2.4. Functions of the PRE and REF commands . . . 9
03. Memory organization: from subchannel to cell . . . 10
3.1. Levels of memory organization . . . 10
3.2. Rank and DRAM device width . . . 10
3.3. Banks and bank groups in a DRAM device . . . 11
3.4. Row size and the limits of independence . . . 12
3.5. Supporting module components . . . 12
3.6. Error correction inside DRAM and channel protection . . . 13
3.7. Key DDR5 differences . . . 13
04. The CK clock signal, MT/s, and nanoseconds . . . 15
4.1. Data rate and the DDR5-6000 designation . . . 15
4.2. Latency comparison for CL30 and CL36 . . . 16
4.3. Minimum intervals in cycles and nanoseconds . . . 16
4.4. The memory clock and the controller clock . . . 16
05. Request processing: CPU, IMC, and scheduler . . . 17
5.1. Conditions for DRAM access . . . 17
5.2. Address mapping and memory resource selection . . . 17
5.3. Scheduler command selection . . . 18
06. Commands and bank states . . . 19
6.1. Basic DRAM operations . . . 19
6.2. Access scenarios for row A . . . 20
6.3. Timing constraints on issuing PRE . . . 21
07. Timings as a system of constraints . . . 22
7.1. Types of timing constraints . . . 22

7.2. Combined effect of timing constraints . . . 22
7.3. BIOS settings and application performance . . . 23
08. Row activation, reading, and closing . . . 24
8.1. tCL: interval from READ to the start of data output . . . 24
8.2. tRCD: interval from ACT to READ or WRITE . . . 24
8.3. tRP: bank preparation for the next activation . . . 25
8.4. tRAS: minimum row active time . . . 25
8.5. tRC: the full cycle of one bank . . . 25
09. Activations: RRD and the sliding FAW window . . . 27
9.1. Constraints on issuing ACT . . . 27
9.2. Four activations in a sliding window . . . 27
9.3. Relationship between FAW and 4 × RRD . . . 28
9.4. Scope of the FAW window and additional events . . . 29
10. Column access and CCD intervals . . . 30
10.1. Data transferred by one access command . . . 30
10.2. First-response latency and continuous data transfer . . . 31
10.3. tCCD_L_WR and tCCD_L_WR2: write sequence requirements . . . 31
10.4. Bank group interleaving . . . 32
10.5. Error correction, ECS, and CRC transfer checking . . . 32
11. Writes, data direction changes, and PRE eligibility . . . 34
11.1. A shared data bus and two signal sources . . . 34
11.2. CWL: interval from WRITE to the start of data transfer . . . 34
11.3. tWTR: interval from write data completion to READ . . . 35
11.4. READ → WRITE: data direction change . . . 35
11.5. tWR and tRTP: intervals before issuing PRE . . . 36
11.6. The tCCD_WTRA constraint . . . 37
12. Refresh and data retention . . . 39
12.1. Periodic refresh of DRAM cells . . . 39
12.2. tREFI and tRFC: maximum average interval and minimum pause . . . 39
12.3. Normal refresh mode and FGR . . . 40
12.4. REFsb: refreshing banks with the same number . . . 41
12.5. Refresh scheduling and postponement limits . . . 42
12.6. Temperature, RFM, and PRAC . . . 43
13. DRAM maintenance modes . . . 44
13.1. Power-down: entry, duration, and exit constraints . . . 44
13.2. Self-refresh: autonomous data refresh . . . 44

13.3. Mode registers: MRR and MRW . . . 46
13.4. ZQ calibration and the DQS oscillator . . . 46
14. Parallelism and command scheduling . . . 47
14.1. Two banks on one timeline . . . 47
14.2. Interleaving, locality, and the request queue . . . 47
14.3. Rank and subchannel parallelism . . . 48
15. Latency, bandwidth, and workload . . . 49
15.1. CAS latency and total request latency . . . 49
15.2. Interface bandwidth . . . 49
15.3. Memory parameters on the critical path of a request . . . 50
15.4. Test result interpretation . . . 50
16. Intel memory controller: tertiary timings, RTL, and refresh parameters . . . 52
16.1. RDRD, WRWR, RDWR, and WRRD notation . . . 52
16.2. RTL, QCLK, and memory training . . . 52
16.3. Controller refresh parameters . . . 53
17. Summary model of memory operation and practical application . . . 54
17.1. A unified model of the operation sequence . . . 54
17.2. Five criteria for timing analysis . . . 55
17.3. Further reading . . . 55
Appendix A. DDR3 → DDR4 → DDR5: evolution of memory architecture . . . 56
A.1. Shared principles and differences across DDR generations . . . 56
A.2. DDR3 → DDR4: architectural changes with 8n prefetch retained . . . 56
A.3. DDR4 → DDR5: transition to two subchannels . . . 57
A.4. BL16 transfer of 64 bytes over 32 data lines . . . 58
A.5. Timing names and differences in their conditions of use . . . 59
A.6. Changes in interbank constraints and post-write intervals . . . 60
A.7. CL8, CL16, and CL30: examples of 10 ns CAS latency . . . 61
A.8. Four measures of time and transfer rate . . . 62
A.9. Refresh organization . . . 62
A.10. Power, data protection, and interface training . . . 62
A.11. The effect of architecture on memory performance . . . 63
Appendix B. Abbreviations and notation . . . 64
B.1. Abbreviations, commands, and registers . . . 64
B.2. Timing parameters . . . 67
B.3. Units of measurement and notation . . . 69
Sources and Documentation . . . 70

01. Introduction to DDR5 theory

The processor runs a program, while main memory holds the data it needs. Let's examine how memory receives a request, finds the data, and sends it back—and why this takes time.

1.1. Storage locations for a running program's data

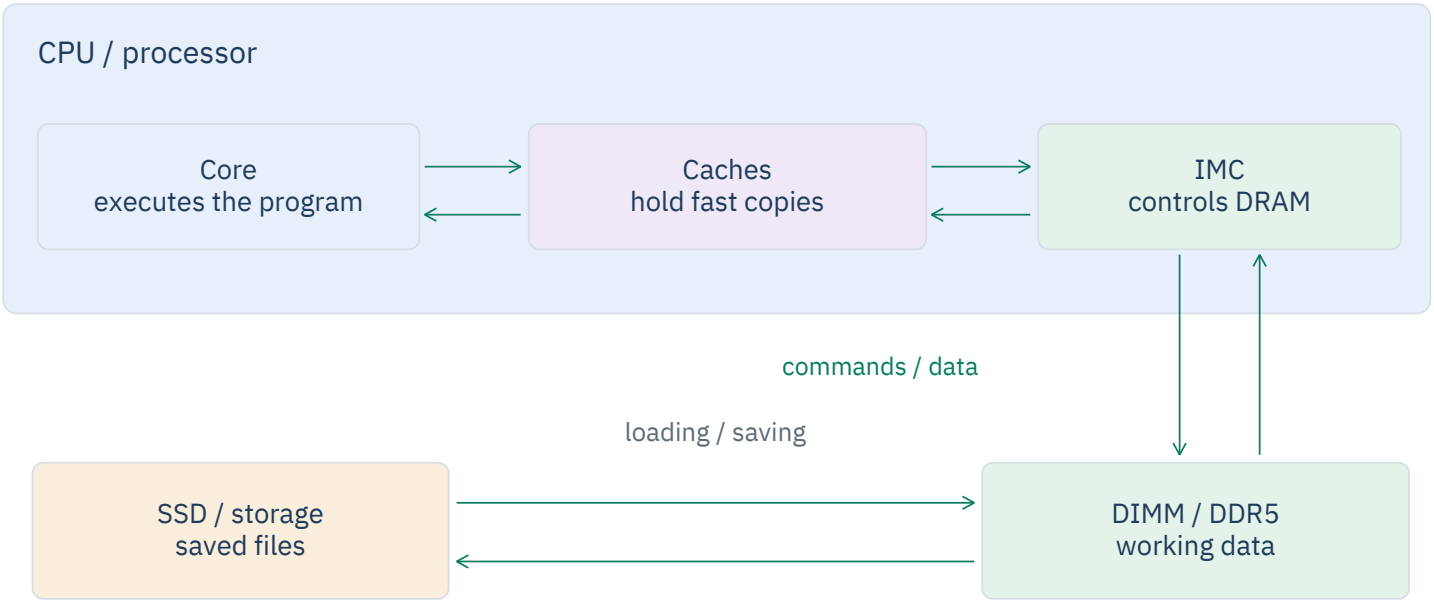
When you open a photograph, launch a game, or perform calculations in a spreadsheet, the processor (CPU) executes the program's instructions and works with its data. Random-access memory (RAM) holds the instructions and data currently in use. DDR5 is one generation of this memory.

Files reside on a storage device, such as an SSD. During use, the required parts of those files are loaded into RAM. To retain data, ordinary DRAM needs power and regular maintenance. Its contents do not survive when the computer is turned off, so you need to save your work to a file.

The processor contains caches: small, fast stores of data copies. If the required copy is already there, no DRAM access is needed.

The memory controller manages communication: it receives requests, selects the order in which to serve them, and sends commands to DRAM. In the modern platforms considered here, it is built into the processor and is called the IMC. A DIMM is the familiar memory circuit board that plugs into a computer's memory slot.

Figure 1. Where data resides and how it moves



Transfers to and from DDR5 pass through the IMC. File loading and saving are shown as logical transfers; I/O interfaces are not shown.

1.2. Bits, bytes, and basic memory characteristics

A bit is a unit of information with two possible values: 0 or 1. Eight bits make a byte. Programs encode numbers, letters, and images as sequences of bits. A memory chip stores these sequences in cells; we will examine DRAM cells in chapter 2.

Capacity tells you how much data memory can hold. Latency tells you how long you must wait for a response to a request. Bandwidth tells you how much data can be transferred per second.

Prefixes are used for large quantities. Each successive decimal unit is a thousand times larger than the previous one: 1 kB = 1000 bytes, 1 MB = 1000 kB, 1 GB = 1000 MB = 1 000 000 000 bytes. For binary units, the multiplier is 1024, or 2^{10}. These units are useful for describing memory with binary addressing and have their own names.

A kibibyte (KiB) is 1024 bytes. A mebibyte (MiB) is 1024 KiB, or 1 048 576 bytes. A gibibyte (GiB) is 1024 MiB, or 1 073 741 824 bytes. Their international symbols are KiB, MiB, and GiB. The inserted syllable “bi” identifies a binary prefix. For example, 4 KiB contains 4096 bytes, while 4 kB contains 4000 bytes; 1 GiB is approximately 1.074 GB.

In English notation, bit or b means a bit, while B means a byte. This manual uses Gbit for die density. Document titles retain the manufacturer's notation, including Gb. We use KiB and GiB for exact binary sizes; for transfer speed, GB/s means a billion bytes per second.

In DRAM documentation, density notation is historically tied to the binary organization of the array. For example, the SK hynix die considered here, with a specified density of 32 Gbit, contains 32×2^{30} bits of user data, or 4 GiB. Density describes one die; a kit's capacity is the combined capacity of all its modules. We will examine module organization in chapter 3.

A nanosecond, ns, is one billionth of a second; a microsecond, µs, is one millionth. Interface operation is also measured in clock cycles. A clock cycle is one period of a repeating synchronization signal. In chapter 4, we will convert clock cycles to nanoseconds using a numerical example.

Property	Question	Example unit
Capacity	How much data will fit?	GiB
Latency	When will the required response arrive?	ns
Bandwidth	How much data will arrive per second?	GB/s

1.3. Main topics of the guide

The memory array is organized into rows and columns. It is divided into banks: parts with their own state for the selected row. The controller first prepares the required row, then selects a portion of data to read or write. These operations take different amounts of time depending on the bank's initial state.

While one bank prepares a row, another can do its own work. However, several banks may use the same data transmission lines. Such shared lines are called a bus. The wait for an individual response and the rate of a large stream of requests therefore depend on different constraints.

A timing constraint specifies a time requirement that must be met before the next action. With a lower timing value, the controller can shorten the interval if the device permits that operating mode. This does not necessarily make the program run faster: a queue, a busy bus, or another requirement may still delay the response.

> The critical path is the chain of waits that determines when a particular result becomes available. Changing a parameter speeds up that request only if it shortens a wait on this path.

1.4. Guide scope and reading sequence

The main command model applies to DDR5 SDRAM: synchronous memory that stores data dynamically. The basic teaching examples use BL16, a burst of 16 transfers on each data line. Whenever a conclusion depends on a different x4/x8/x16 organization, density, or special mode, we state it alongside that conclusion.

The detailed numerical requirements for the selected device come from SK hynix 32Gb DDR5 SDRAM, Rev. 1.2, July 2025. Chapter 16 separately examines controller fields from the specified Intel documents. These levels of description must be distinguished when comparing numbers.

Chapters 2–6 provide the foundation: the cell, module, clock cycle, request, and commands. Chapters 7–13 explain constraints, refresh, and maintenance modes; chapters 14–15 cover parallelism and performance. The main route ends with the overview map in chapter 17. You can leave chapter 16 and appendices A and B for a second reading: they collect platform fields, a detailed comparison of generations, and a reference guide to notation.

A datasheet is a manufacturer's technical description of a device. BIOS/UEFI is the motherboard's built-in software, whose menus provide access to settings. A BIOS field name may differ from the notation for a DRAM constraint. Technical terms appear alongside their explanations.

02. The DRAM cell, sensing, and charge restoration

DRAM stores a bit as an electrical state. To read it reliably, memory amplifies a weak signal and restores the cell's contents.

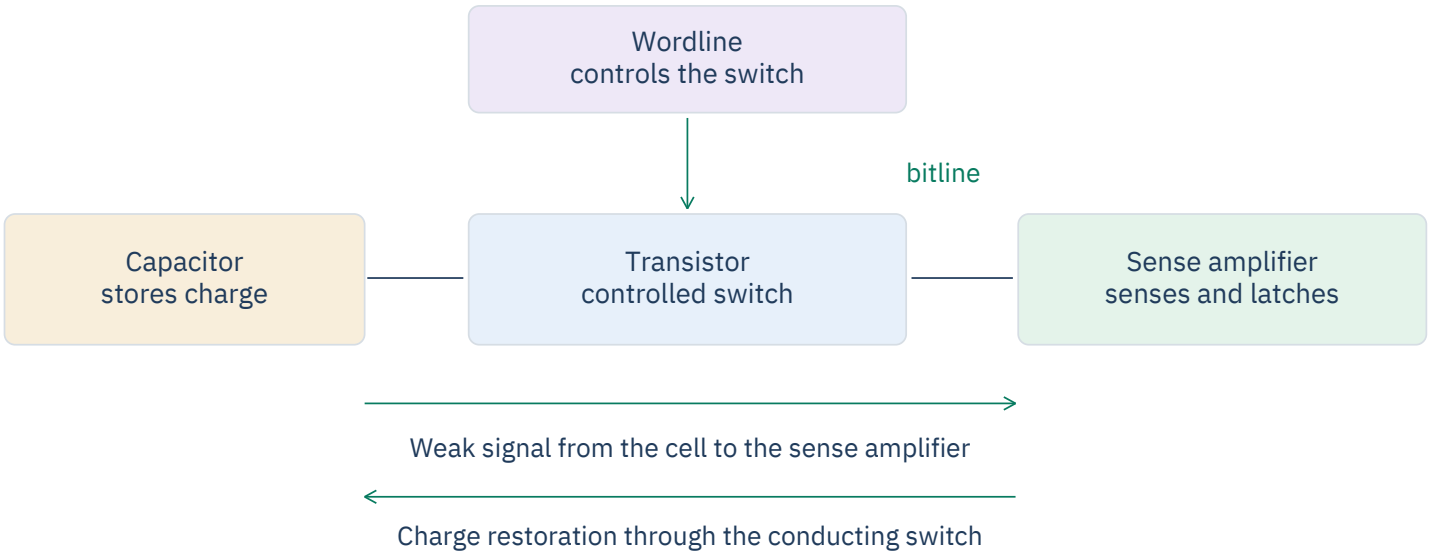
2.1. Charge, a switch, and array lines

A simplified DRAM cell consists of a capacitor and an access transistor. A capacitor is a component that can hold electric charge. The transistor acts as a controlled switch: it connects the cell to the bitline or disconnects them. Two distinguishable charge states encode one bit. This model is called 1T1C: one transistor and one capacitor.

The wordline controls the transistors of the cells in a row. The bitline is the line through which the selected cell's state affects the input of the sense amplifier. The words row and column refer to objects selected at different stages of an access: first a row in the array is selected, then the required part of the active row is selected for transfer.

Charge leaks away over time. Even if the program changes nothing, the states become harder to distinguish. DRAM therefore needs periodic data restoration. This is the reason for the D: dynamic, referring to dynamic data storage.

Figure 2. The cell and array interconnects



The 1T1C model: the wordline controls the switch, and the bitline connects the cell to the sense amplifier.

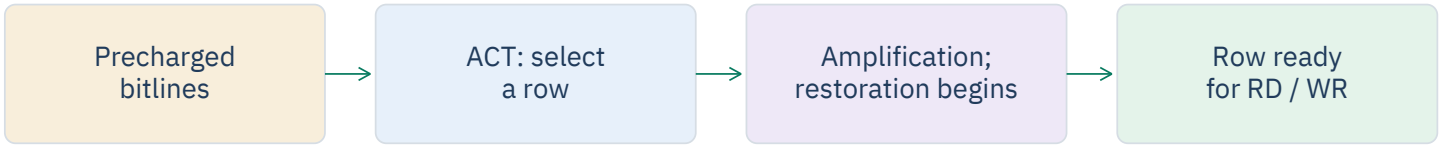
2.2. Row activation

Before activation, the bitlines are prepared for sensing. When the selected cell's transistor turns on, charge redistributes between the cell's small capacitance and the bitline. This creates a weak voltage deviation. The sense amplifier detects this deviation, amplifies it to a stable logic state, and holds the result.

This internal sensing changes the cell's original charge state, which is why it is called destructive sensing. Even so, data must be preserved after a normally completed operation: while the cell remains connected to the amplifier, its charge is restored to a reliable level. Amplification and restoration take place during activation.

The signal must have time to become reliable enough for the next action. Outputting the selected data and safely completing the row cycle require different preparation times. This distinction will later explain why both tRCD and tRAS exist.

Figure 3. Activation and internal sensing



A small charge change becomes a stable sense-amplifier state.

RD later selects part of the active row for external transfer.

A sequence of states, with no time scale. Row restoration can continue after RD becomes permissible; tRAS is checked separately.

2.3. The open row and row buffer

An open row means that the bank is in its active state: its sense amplifiers hold the data from the selected row. This set of amplifiers is commonly called the row buffer.

While the row is open, its cells remain connected to the amplifiers. The amplifiers hold the sensed values and support restoration of cell charge. Closing the row breaks this connection and prepares the bitlines for new sensing. In the standard command model, one row is active in a bank.

ACT activates a row inside DRAM: the row data remains in the sense amplifiers. RD is issued later and selects part of the already active row for transfer through the external data pins. Cell charge is sensed during activation; subsequent RD commands access the data in the row buffer.

2.4. Functions of the PRE and REF commands

Before a different row can be opened in the same bank, the bank must return to its prepared state. PRECHARGE, abbreviated PRE, ends work with the active row and prepares the bitlines for the next activation. The requirements for charge restoration and data writing must be met before the row is closed. After PRE, the tRP interval must elapse; only then is the bank ready for the next ACT.

Normal activation restores the row being accessed. However, a program need not access every row regularly. This is why refresh is needed: memory periodically senses and restores row data independently of program accesses. This keeps cell states distinguishable as charge leaks away. We will examine the refresh schedule and the resources it occupies in chapter 12.

03. Memory organization: from subchannel to cell

Banks, ranks, and subchannels allow some operations to run in parallel. Each level, however, has its own shared resources and constraints.

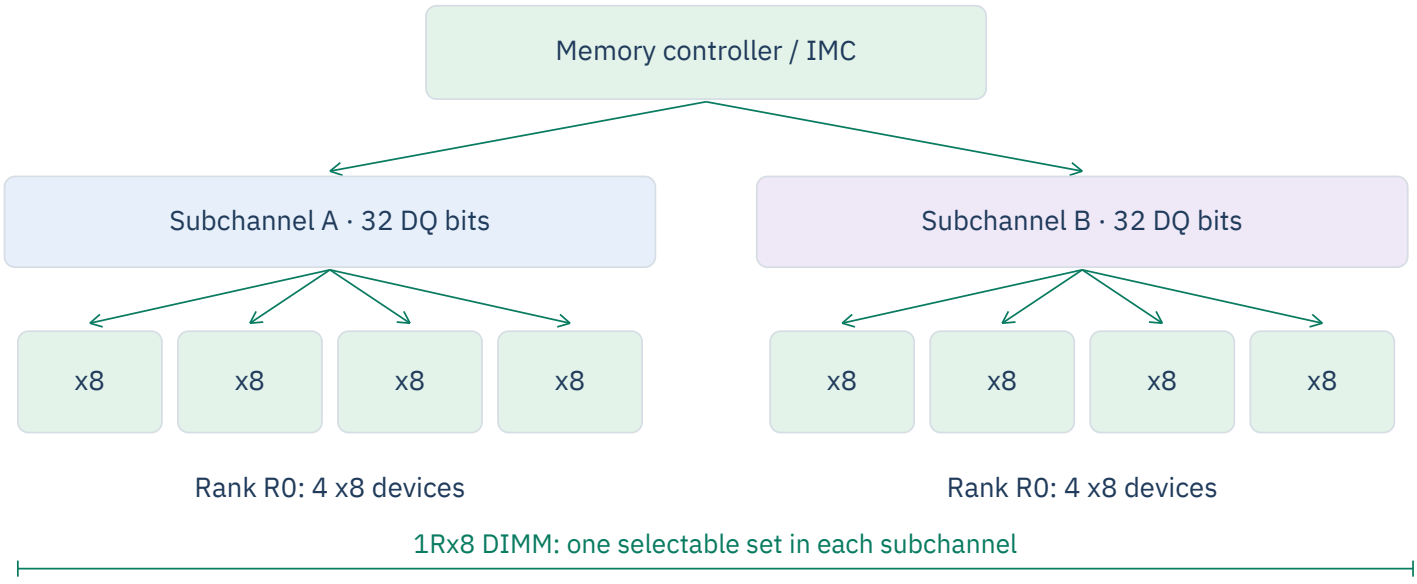
3.1. Levels of memory organization

A DIMM is a memory module with ICs and supporting components. A DRAM device is an individual memory chip with its own interface and internal organization. A channel connects several such devices. The packages on the board do not always reveal the number of dies: one package can contain several dies.

In a typical DDR5 DIMM, the 64 data bits of the interface are divided into two 32-bit subchannels. Each subchannel can receive its own sequence of commands and transfer its own data. Modules with additional error-correction bits have a wider physical interface; here, 32 bits refers specifically to data, excluding those additional lines.

Manufacturers and software use the word channel at different levels: for a platform's 64-bit interface, its 32-bit DDR5 subchannel, or a separate controller entity. It is therefore more useful to establish the width and command independence first, and only then compare the “dual channel” labels in two programs. The two subchannels are within one DDR5 module; two independent memory channels are at the processor's memory-controller level.

Figure 4. Subchannels and devices working together



An example of a complete 1R DDR5 DIMM without external ECC bits, using x8 devices. The connections shown are logical.

3.2. Rank and DRAM device width

A rank is a set of devices selected together. They receive coordinated commands and transfer different data bits. UDIMM means a module without a register that buffers commands and addresses.

In an ordinary 1Rx8 UDIMM without external ECC, one selectable set of four x8 devices operates on each of the two subchannels. A 2Rx8 module has two such sets on each subchannel. The sets on one subchannel share its data lines. A second rank adds memory arrays; the number of external subchannels remains two.

The designation x8 means that one device has eight data lines. To provide the subchannel's 32 data bits, four x8 devices work together in a simple organization without external ECC. Eight x4 devices or two x16 devices would be needed instead.

A rank and a side of the printed circuit board are different concepts. The ICs of one rank may be located on both sides; packages may contain more than one die. You cannot reliably replace 1R/2R with "single-sided/double-sided" or determine the organization solely from the number of visible black packages.

Object	What it represents	What it cannot tell you
DIMM	The module and its connections	The number of ranks from the placement of ICs on the board
Rank	A set of devices selected together	An independent data bus for each rank
x4 / x8 / x16	The DQ interface width of one device	Capacity and bank count without a datasheet
Subchannel	A domain for commands and data transfers	Complete independence from the IMC and power supply

Capacity is conveniently calculated for the whole module. In a complete DIMM using monolithic x8 devices without external ECC, one rank comprises eight devices: four on each subchannel. Under the accepted binary density convention, 16 Gbit gives 2 GiB per device, while 32 Gbit gives 4 GiB.

Organization	Devices in the entire DIMM	Device density	DIMM capacity
1Rx8	8	16 Gbit	16 GiB
1Rx8	8	32 Gbit	32 GiB
2Rx8	16	16 Gbit	32 GiB

These are arithmetic examples of organization, unrelated to any particular kit. The same 32 GiB can come from eight 32-Gbit devices or sixteen 16-Gbit devices. This difference matters for refresh: device parameters depend on die density, as we will examine in chapter 12.

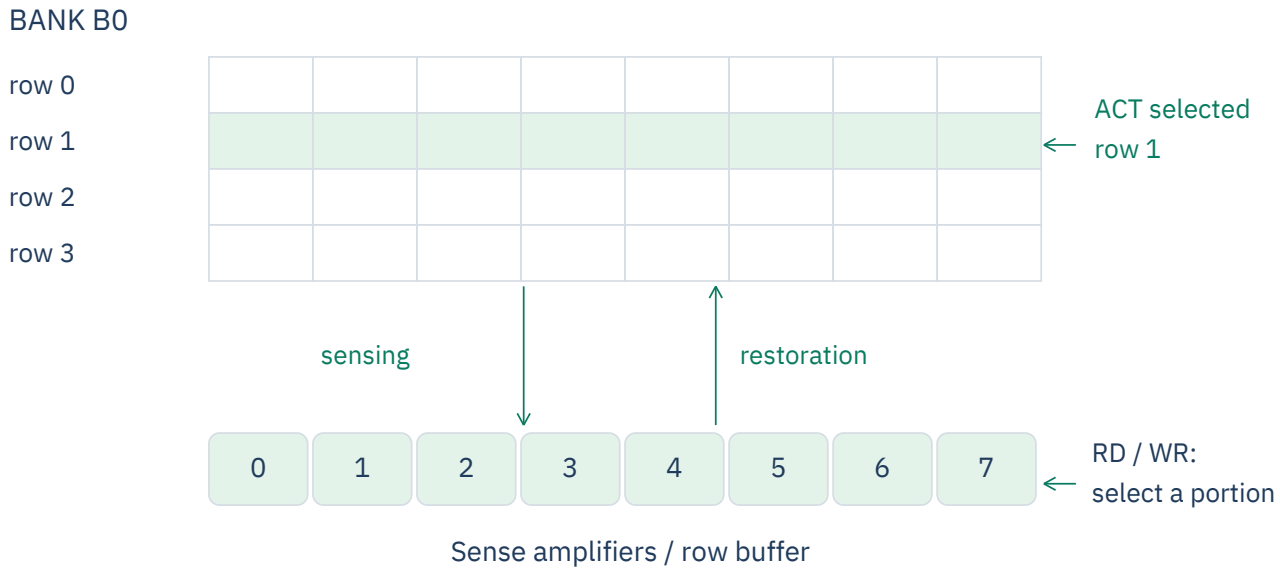
3.3. Banks and bank groups in a DRAM device

Inside DRAM, the array is divided into banks. Each bank has its own selected active row, so some operations in different banks can take place simultaneously. A bank group, or BG, combines banks subject to additional shared constraints. Some commands to banks in the same group require a longer interval between them than commands to banks in different groups.

For densities of 16–64 Gbit, the DDR5 x4/x8 organization considered here uses eight groups of four banks, or 32 banks per device. The x16 organization considered here has four groups of four, or 16 banks. For 8 Gbit, Table 1 of the same SK hynix datasheet lists two banks per BG: 16 banks for x4/x8 and 8 for x16. Always check a particular device's geometry and density in its datasheet.

Rows and columns are inside a bank. The row address selects a row during ACT; the column address helps select a portion of data during RD or WR. The physical array is more complex and contains subarrays. Here, the bank model used by the external controller when issuing commands is sufficient.

Figure 5. Rows, columns, and the row buffer



The selected row stays connected until it is closed.

One logical bank of a device. The selected row is connected to sense amplifiers that sense the data and restore charge. RD/WR commands select a portion of the data.

3.4. Row size and the limits of independence

The row size of a single DRAM device is the amount of data accessible after activation; geometry tables also call it the page size. In the datasheet considered here, x4 has 2048 columns × 4 bits = 1 KiB, x8 has 1024 × 8 bits = 1 KiB, and x16 has 1024 × 16 bits = 2 KiB. Four x8 devices operating in parallel provide a 4 KiB active row for the subchannel. A read command outputs a selected portion of that row as one burst.

If four x8 devices serve a subchannel in parallel, their corresponding banks are activated in coordination, and their data bits are combined. A command to the rank addresses all participating devices at once. Multiplying the number of devices by the number of banks therefore does not give the number of banks that the controller can manage independently.

Another bank can prepare a row while the first transfers data. Another rank has its own arrays and states, but usually uses the same data lines on its subchannel. Another subchannel removes some of this contention, but the controller, processor queues, power supply, and platform constraints remain shared. The degree of parallelism depends on the resource in question.

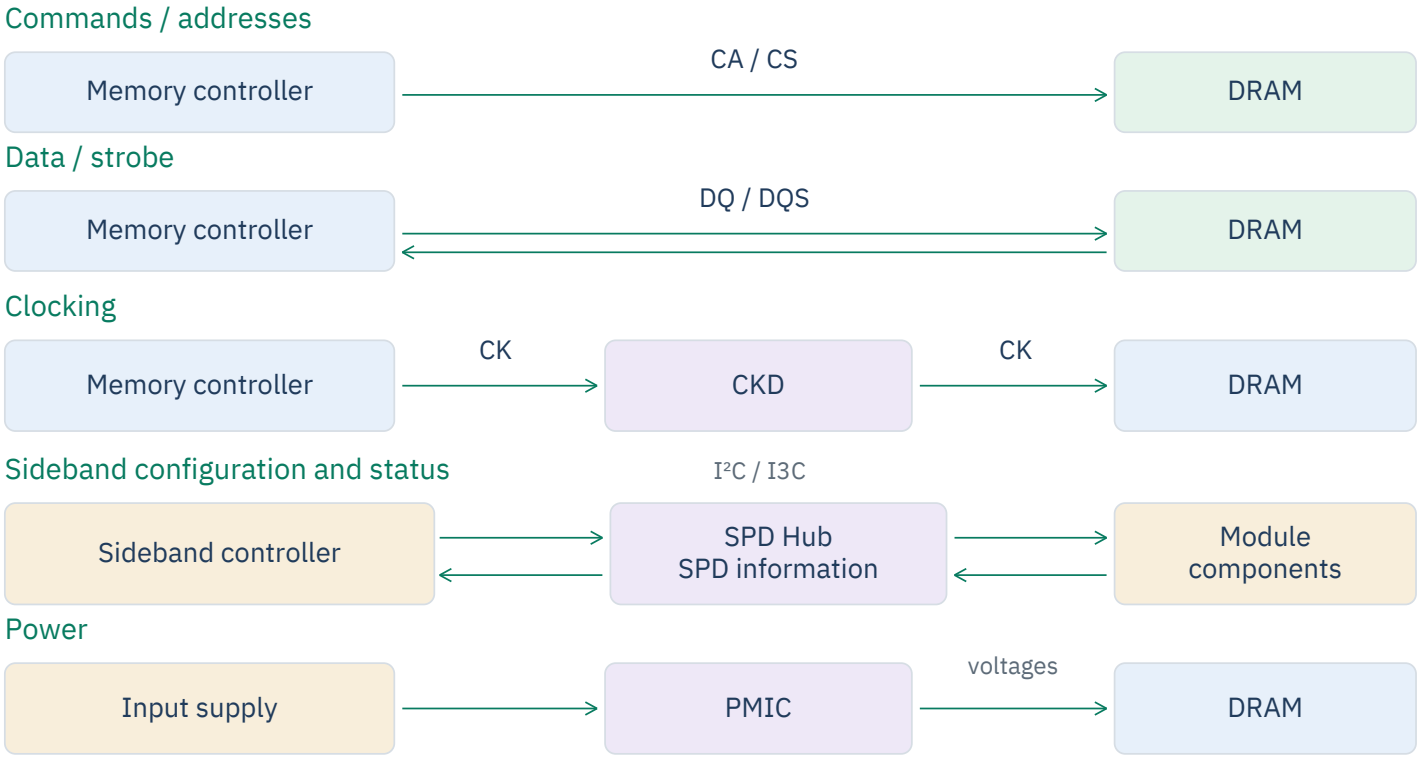
3.5. Supporting module components

SPD, Serial Presence Detect, stores information about the module in nonvolatile memory. In DDR5, the SPD Hub connects the external I²C/I3C Basic management bus to local components, allowing the system to obtain their parameters and status. I²C is a serial bus with two signal lines: data and clock. I3C Basic also uses two lines, but supports faster management communication and in-band notifications from the device. In the implementation considered here, the hub also contains a temperature sensor.

The PMIC, Power Management Integrated Circuit, generates the voltages for the module's ICs from the input supply and controls their delivery.

CUDIMM and CSODIMM are versions of the standard and compact modules with a CKD, Client Clock Driver, clock buffer. It buffers and distributes CK from the controller to DRAM. Its modes and compatibility depend on the platform.

Figure 6. A module has several separate connection paths



Each row shows a separate function of a client DIMM, so the DRAM block is repeated. A CKD buffers CK on CUDIMMs/CSODIMMs; on a regular UDIMM, the clock reaches DRAM without a CKD.

3.6. Error correction inside DRAM and channel protection

On-die ECC, or ODECC, is built-in error correction that corrects single-bit errors inside the DRAM die. It does not correct errors that arise along the external data transfer path to the controller. System ECC protection requires controller support and an appropriate memory organization with additional check data. In section 10.5, we will examine internal correction, ECS maintenance (checking, correcting, and storing the result), and transfer checking with CRC (a cyclic redundancy code).

3.7. Key DDR5 differences

The main differences are apparent in internal prefetch size and transfer organization. Prefetch is the portion of data that DRAM prepares in parallel for subsequent output: 8n and 16n mean 8 and 16 bits per DQ line. BL specifies the number of transfers in an external burst. The comparison below covers specific x8 devices and ordinary UDIMMs without system ECC; the details of each generation are retained in appendix A.

Feature	DDR3, 2 Gbit x8	DDR4, 8 Gbit x8	DDR5, 32 Gbit x8
Internal prefetch	8n	8n	16n
Device banks	8, no groups	4 BG × 4 = 16	8 BG × 4 = 32
DIMM data interface	64 DQ	64 DQ	2 subchannels × 32 DQ
Basic full burst	BL8	BL8	BL16

Two subchannels allow different accesses to be served, while BL16 preserves a 64-byte portion at a width of 32 DQ. The bank counts apply to the specified densities and organization; x4/x16 variants and other burst modes are covered in appendix A.

04. The CK clock signal, MT/s, and nanoseconds

The same number of clock cycles gives a different delay in nanoseconds at different memory data rates. Comparing timing values requires the duration of one CK cycle.

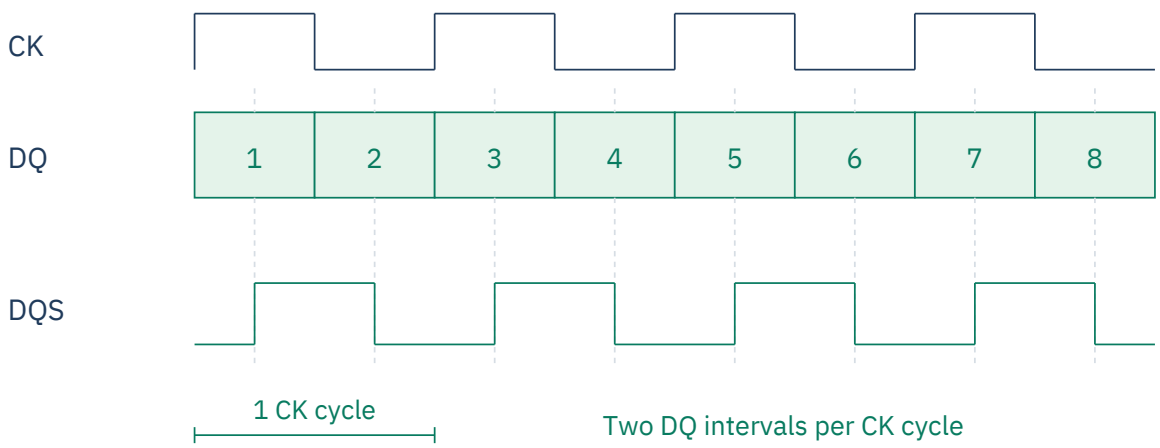
4.1. Data rate and the DDR5-6000 designation

DDR means double data rate: data is transferred twice per clock period. For an interface with a nominal data rate of 6000 MT/s, the CK clock frequency is 3000 MHz. MT/s means millions of transfers per second on one data line; MHz means millions of periods per second.

CK is the interface clock signal, and tCK is its period. The notation nCK means a number of these periods. Processing inside the array may run at a different frequency from the CK clock: DDR5 uses internal parallel prefetch and serializes data on the external interface. The controller frequency can also differ from the CK frequency.

DQ are the lines that carry data. DQS is the accompanying strobe: its edges tell the receiver when to capture data. The controller sends DQ and DQS during a write; DRAM sends them during a read.

Figure 7. CK, DQ data, and the DQS strobe during a write



1–8 are transfer numbers; the figure shows part of BL16.

A schematic write fragment: the controller drives DQ and DQS, with DQS centered in the data windows at the DRAM input. The actual phase relative to CK, preamble, postamble, and tolerances are not shown. During a read, DRAM is the source, and DQS is aligned with DQ boundaries at the output; the receiver adjusts the capture point.

At the DRAM input during a write, DQS edges lie within valid data windows; during a read, the initial alignment of DQS and DQ is different, and the receiver adjusts the capture point.

DQS has special segments immediately before and after the data burst: the preamble and postamble.

tCK, ns = 2000 / data rate, MT/s
interval, ns = nCK × tCK

4.2. Latency comparison for CL30 and CL36

The CL number alone does not determine latency. In the selected teaching examples, CL30 at 6000 MT/s corresponds to 10 ns, while CL36 at 8000 MT/s corresponds to 9 ns; the calculations are collected in the table.

This accounts only for the READ→start of data interval. It does not include queuing, opening a new row, the request path from the core, or the return path through the processor.

BL16 means 16 transfers per DQ line, two per CK cycle: the burst occupies 8 CK. At 6000 MT/s this is 2.667 ns; at 8000 MT/s it is 2 ns. CL specifies the time until data starts, while BL specifies the number of transfers.

Example configuration	CK clock frequency	tCK (period)	CL × tCK	BL16 / 2
6000 MT/s, CL30	3000 MHz	0.333… ns	10 ns	8 CK cycles = 2.667 ns
8000 MT/s, CL36	4000 MHz	0.25 ns	9 ns	8 CK cycles = 2 ns

4.3. Minimum intervals in cycles and nanoseconds

The minima in clock cycles and nanoseconds must both be met. For example, suppose an interval must be at least 8 CK and 5 ns. At 6000 MT/s, one cycle takes 1/3 ns, so 5 ns requires 15 CK. This is greater than 8 CK: the effective minimum is 15 CK.

Round the minimum interval upward to a permitted increment. The resulting number must also be checked against the register encoding or available BIOS values: some fields do not permit every whole cycle. For a maximum permitted interval, such as one imposed by a refresh-frequency requirement, the meaning of the condition and the safe direction of rounding will be different.

Minimum in CK cycles = max(limit in nCK, ceil(limit in ns / tCK))

> nCK is the number of cycles; tCK is the cycle duration. max selects the largest number, and ceil rounds upward to an integer. At 6600 MT/s: 5 ns / tCK = 16.5; with an increment of 1 CK, the minimum is 17 CK. Register encoding and operating-mode conditions are checked separately.

4.4. The memory clock and the controller clock

QCLK is the controller clock in Intel documentation. Gear specifies the frequency ratio between the physical memory interface (DDR PHY) and the controller. In Gear 2, the interface frequency is twice as high, so tQCLK = 2 × tCK.

At 6000 MT/s, tCK = 1/3 ns and tQCLK = 2 × tCK = 2/3 ns. The same number of CK and QCLK cycles therefore represents different amounts of time. In the Intel MC_BIOS_REQ register, 5E00, the GEAR field selects the mode; the document used here is ID 767624, Rev. 1.3.

Command Rate describes command issue and reception. Neither it nor Gear can be added to CL or substituted for tCK. The unit of each field must be established from the documentation: for example, RTL in QCLK is examined in detail in section 16.2.

05. Request processing: CPU, IMC, and scheduler

The CPU requests data at an address. The controller turns that request into a valid sequence of memory commands.

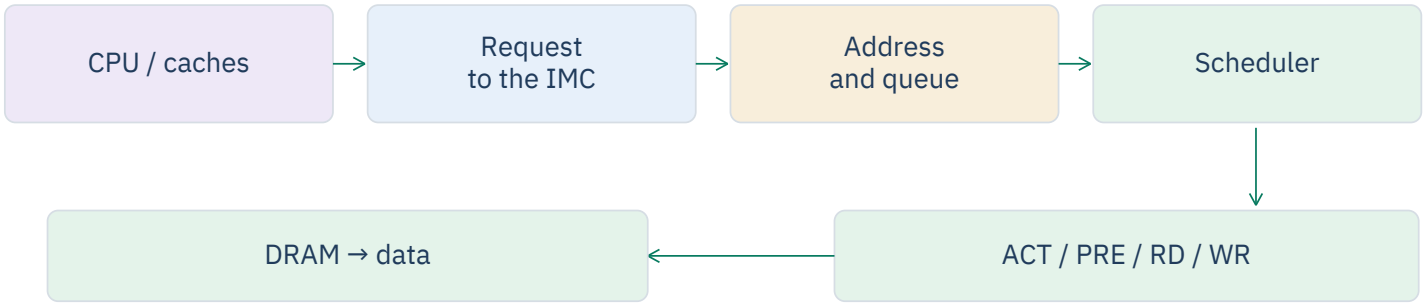
5.1. Conditions for DRAM access

Data access begins with the processor caches. After a miss at one cache level, the search continues: the data may be found farther down the hierarchy or supplied by another cache through the coherence protocol.

Let's take a request that needs DRAM. It travels through the processor's internal interconnects and memory-subsystem queues to the controller. The part of the processor outside its execution cores is called the uncore. The organization of this path depends on the CPU model; a ring interconnect is one possible design.

If an instruction uses the value being read, it must wait for the result. Writes often accumulate in buffers and are sent later. The completion of a store instruction, a WR command on the interface, and the physical restoration of the written cells are therefore three different events.

Figure 8. From a request to data transfer



The logical sequence for a request that needs DRAM.

5.2. Address mapping and memory resource selection

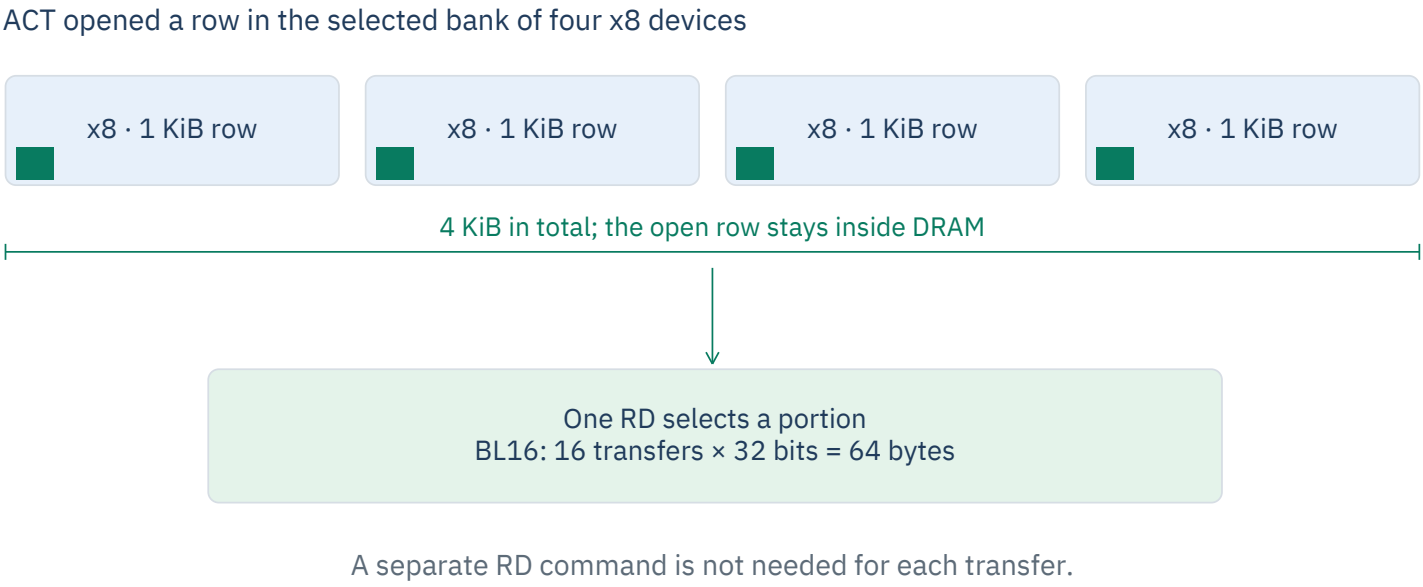
The program's virtual address is first translated into a physical address. The IMC uses the request's physical address to determine where the data resides: channel or subchannel, rank, bank group, bank, row, and column. This correspondence is called address mapping.

The controller may mix address bits and distribute accesses to increase parallelism. Two consecutive physical addresses therefore do not necessarily correspond to two consecutive columns in one bank on every system.

The exact mapping depends on the platform, configuration, and settings; establishing it requires information about the particular implementation. To examine commands, its result is sufficient: for example, rank 0, BG1, bank 2, row A.

After selecting the bank and row, the controller includes the part of the address that determines the column in RD or WR. One such command starts an entire data burst: a separate command is not needed for each of the 16 transfers in BL16. Across 32 data lines, BL16 transfers 16 × 32 bits, or 64 bytes; this is the portion of the active row selected for external transfer.

Figure 9. An active row is larger than one data burst



Illustrative example: four x8 devices in one subchannel; BL16 over 32 payload DQ lines. Active row: 4 × 1 KiB = 4 KiB; burst: 16 × 32 bits = 64 bytes. The highlighted portion is schematic; offset order and address mapping are explained in the text.

In depth: data order within a burst

Data order within a burst is also governed by DRAM rules. For example, in SK hynix Table 29, READ BL16 with C3 = C2 = 0 uses offsets 0–15, while with C3 = 0 and C2 = 1, the order starts with 4–7, followed by 0–3, 12–15, and 8–11. These are local offsets within the selected portion. They cannot be interpreted directly as the low-order bits of an arbitrary program address: the controller's address mapping applies first. In chapter 10, we will distinguish this portion from internal parallel prefetch.

5.3. Scheduler command selection

The scheduler selects a valid command based on request dependencies, bank states, bus occupancy, read/write queues, and refresh deadlines. Commands may be issued in a different order from the requests, but the behavior visible to the program preserves the established rules.

Prioritizing an open row saves closing it and activating a new one, but can delay requests to other rows. The scheduler balances bandwidth, latency, and fairness. The policy of a particular commercial IMC cannot be inferred from the general model.

Configuration parameters specify constraints, and the controller applies them when choosing when to issue ACT, RD, WR, and other commands. Some settings are also used to program registers in DRAM itself.

> Row locality means repeated access to data in an already active row. Bank-level parallelism is the ability to serve different banks with overlapping operations.

06. Commands and bank states

The bank's state determines which command can be issued next and how long the wait will be. ACT opens a row, RD and WR access its data, and PRE closes the row.

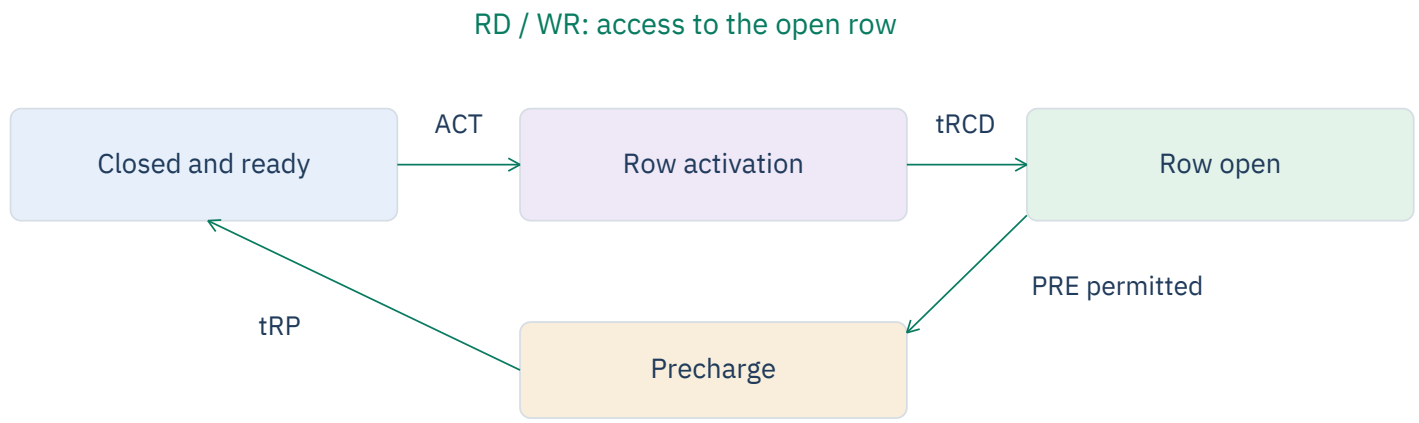
6.1. Basic DRAM operations

ACTIVATE, ACT, selects a row in the specified bank and starts its activation. For an ordinary new access, the bank must be closed and ready. After ACT, the row-to-column delay—the interval from row activation to column access—must elapse before RD or WR can be issued.

READ, RD, selects data from the active row. WRITE, WR, specifies where to write in the active row; the data itself arrives later on DQ, according to the write latency. PRECHARGE, PRE, ends work with the active row and prepares the bank for a new cycle. REFRESH, REF, starts the refresh operation specified by the mode. We will need the more specific REFab and REFsb designations in chapter 12.

Commands and addresses travel on the CA (command/address) bus, while data travels separately on DQ. ACT carries a row address and starts internal row activation. READ initiates data output. RD and WR use the column address and data-burst parameters. PRE changes the bank state and does not transfer data. READ first specifies the bank and group selection, then the part of the address that determines the column.

Figure 10. States of one bank



A simplified model without PD/SR. Each transition is subject to its timing constraints; with auto-precharge, the bank closes according to the mode rules.

Command	Prerequisite	What it does	Main constraints
ACT	The bank is ready for activation	Selects and activates a row	tRP, tRC, tRRD, tFAW
RD	The required row is active	Starts reading the selected portion	tRCD, RD→RD interval, CL
WR	The required row is active	Specifies where to write data that will arrive later	tRCD, CWL, WR intervals
PRE	The row can be closed	Closes the row and prepares the bank	tRAS, tRTP, tWR
REF	The mode's requirements are met	Restores the array's contents	tRP and refresh constraints

6.2. Access scenarios for row A

Suppose a request needs row A in bank B0. If A is already open, this is a row hit. No new ACT is required for a read: after meeting the other constraints, the controller issues RD and waits for data transmission to begin. In this situation, tRCD and tRP must not be added again to every read.

If B0 is closed and ready for activation, ACT A is required first, followed by the tRCD interval and RD. We will call this a closed-bank access. The word “closed” is not sufficient if the interval after PRE has not yet elapsed: readiness for ACT must be checked separately.

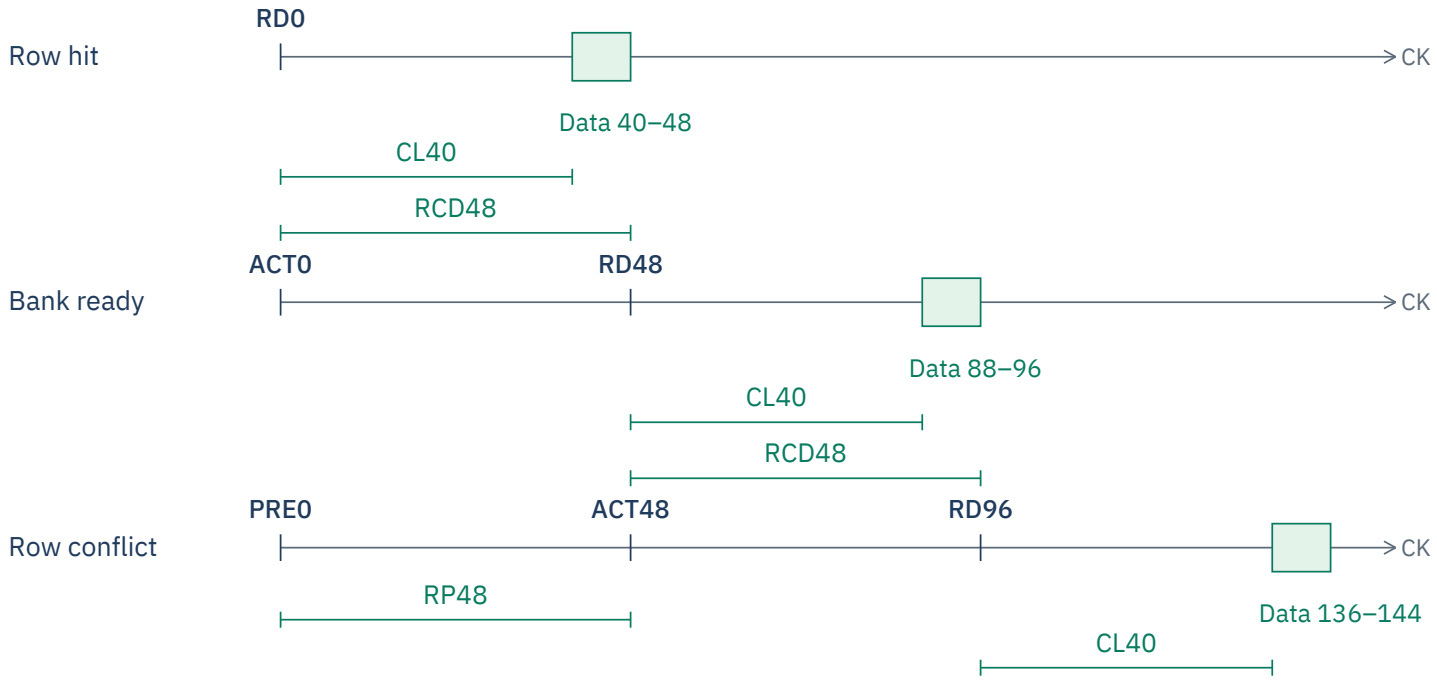
If a different row X is open in B0, this is a row conflict. The controller must wait until PRE can be issued, close X, wait tRP, and open A. From there, the path is the same as for a closed-bank access.

In this guide, a row miss means the absence of a row hit. We distinguish closed-bank access from row conflict: in the first case, no new PRE is needed; in the second, another row must first be closed. Other sources may define row miss differently.

In the teaching diagrams, ACT, RD, and WR are marked at the last cycle of the corresponding two-cycle command, while PRE is marked at its reception cycle. A marker defines the reference point for intervals; the complete command waveform on the CA bus is not shown. CKn denotes the instant n CK cycles after the starting reference. This is why adjoining tRCD and CL intervals add together without an extra +1 CK.

The reference cycles are shown in the SK hynix datasheet: Figures 37–38, p. 152, and Figure 141, p. 279.

Figure 11. Three paths to row A in bank B0



Illustrative example: bank B0 in one rank; BL16; CL = 40, RCD = 48, RP = 48 (CK cycles). PRE is already permitted, and tRC is satisfied for the next ACT; the buses are free. Command markers denote the specified reference points.

6.3. Timing constraints on issuing PRE

Closing an open bank requires several timing constraints to be met. At least tRAS must elapse after ACT, and tRTP applies after READ. If there have been writes, data transfer and the write recovery constrained by tWR must be complete. All requirements of the current access must be met before PRE.

Data from the last read may continue through the external pipeline even after some internal operations have completed. The read constraint runs from RD to PRE; the start of the write-recovery interval is tied to the end of data reception. The reference points are examined in chapter 11.

Auto-precharge, AP, allows automatic closure to be requested after a read or write. The tRAS, write-recovery, and other minimum requirements still apply. The device arranges closure according to the command and mode rules, while the controller must determine when the bank will be available again. RDA/WRA in source diagrams denote these variants.

> The sequences in this chapter show the dependencies of one access when shared resources are available. The full latency of a CPU request additionally includes queues, interface constraints, and the data path inside the processor.

07. Timings as a system of constraints

A timing connects events in memory operation. To read it correctly, you need to know the reference point, the next action, the resource, the state, the mode, and the units.

7.1. Types of timing constraints

In ACT→RD notation, the arrow means “from the ACT command to the RD command.” The letter t in a timing name indicates time; the documentation specifies the exact units. A minimum of 48 CK cycles means that the next action is prohibited until 48 cycles have elapsed. It may occur later, provided the other conditions are met, including the maximum time allowed for maintenance.

Many timings specify a minimum interval between commands, such as PRE→ACT for one bank. CL describes the latency before data output, while tWR is the recovery time after write data has been received. tFAW limits the number of activations in a sliding window, and tREFI sets the scheduling conditions for mandatory array maintenance.

The same pair of commands can have different constraints. ACT→ACT in the same bank is governed by tRC, while ACT→ACT between different banks is governed by the applicable tRRD and additional conditions. For RD→RD, bank placement and the transfer mode matter. When looking at an interval in a diagram, you therefore need to check which banks and mode it applies to.

For the ACT, RD, WR, and PRE markers, we use the convention from section 6.2. In particular, CL starts on the second cycle of READ; the reference cycles are shown in Figures 37–38, p. 152, of the SK hynix datasheet.

Class	Example	What to establish
Command → command	tRCD, tRP, tRTP	Bank, command types, reference cycle
Command → data	CL, CWL	Start of data/DQS, mode
Data → action	tWR, tWTR	End of the write and CK cycle boundary
Sliding window	tFAW	Counting scope and preceding events
Array maintenance	tREFI, tRFC	Refresh mode, duration, and frequency

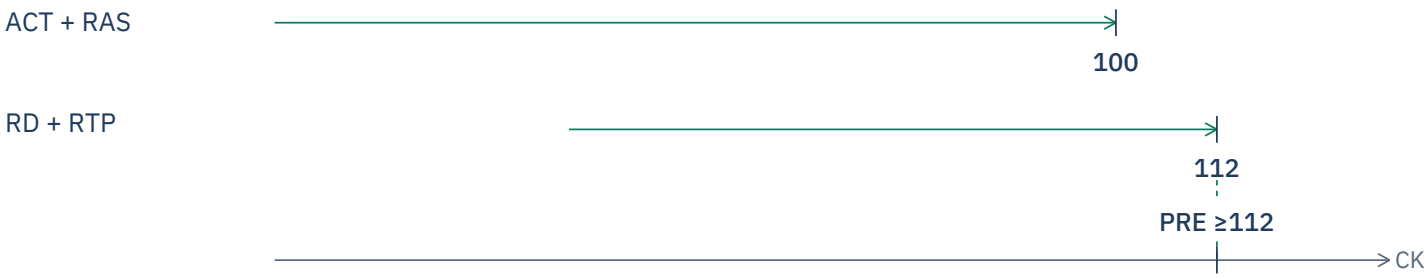
7.2. Combined effect of timing constraints

Suppose tRAS allows PRE for this bank no earlier than cycle 100, while the last READ command and tRTP allow it no earlier than cycle 112. PRE can therefore be issued starting at cycle 112. Both boundaries are on the same timeline; we choose the later one. Reducing the first boundary from 100 to 90 would leave the schedule for this access unchanged.

Addition applies where the stages actually follow one another: PRE, wait for tRP, ACT, wait for tRCD, RD, wait for CL. Even this sum describes only the specified path. Some of these actions can overlap in another bank, and the initial stages are absent for a row hit.

The shared bus may require an even later command issue time. The controller waits until all applicable constraints have been met, and only then does the command become permissible. The scheduler selects a request from the queue; its choice also affects when the command executes.

Figure 12. The latest bound determines when PRE is allowed



Illustrative example for one bank, with time in CK cycles. Independent bounds: ACT + RAS = 100; RD + RTP = 112. PRE is permitted from CK112, the later of the two bounds.

7.3. BIOS settings and application performance

The number entered in the BIOS may differ from the value programmed after training—the automatic adjustment of transfers at startup. The BIOS and the controller initialization code may round it or reconcile it with neighboring fields. On Intel platforms, this code is called MRC.

The programmed value sets one of the scheduling conditions. The effective constraint depends on the entire configuration, while an application's result depends on its memory accesses. A monitoring application's reading alone therefore does not explain the cause of a pause or the size of a gain.

Primary, secondary, and tertiary are the customary groups of BIOS timing fields; their contents depend on the platform. Here, parameters are grouped by action: row access, activations, data transfer, direction changes, refresh, and maintenance transitions.

08. Row activation, reading, and closing

Opening a row, waiting for data, and preparing the bank for a new row are separate stages. Each needs its own interval; the stages in the path depend on the bank's state.

8.1. tCL: interval from READ to the start of data output

CAS latency, CL, specifies the delay from the defined reference point of the READ command to data output in the applicable mode. By the time READ is issued, the row is already prepared. The command selects a portion of data, which then travels along the internal path to the external interface. CL does not include opening the row and does not measure the entire path from the CPU. Figure 34, p. 150, of the SK hynix datasheet illustrates this.

CAS stands for Column Address Strobe. The name is a historical carryover; in DDR5, reading is initiated by the READ command.

After each RD under consideration, CL determines when data transfer begins. Reducing CL can bring the response forward. The interval between adjacent bursts in a continuous stream is also constrained by tCCD, burst duration, and the command schedule, so it may remain unchanged.

The allowed CL values depend on the speed grade, mode, and device encoding. Both data and the strobe must be ready by the start of output. The datasheet specifies the external constraints, while writing uses a separate parameter, CWL/write latency.

> tCL: READ → start of read data output; scope—the selected rank/subchannel and the current read mode. External DQS/DQ have separate electrical tolerances. Related parameters: tRCD, tCCD, burst transfer, and the read-to-write transition.

8.2. tRCD: interval from ACT to READ or WRITE

tRCD, row-to-column delay, limits ACT→READ or ACT→WRITE to the activated row. First, the cell charge affects the bitlines; then the sense amplifiers establish a state suitable for column access. For ACT→WR, the reference cycles are shown in Figure 141, p. 279.

This parameter matters when a new activation is needed: an access to a closed bank or a row conflict. If the required row is already open and the minimum has long since elapsed, each subsequent read does not have to wait for tRCD again. The effect therefore depends on the proportion of accesses that require ACT.

The BIOS may show separate tRCDRD and tRCDWR values for ACT→RD and ACT→WR. These are platform settings; the datasheet specifies the DRAM's minimum requirements. The mapping of these fields to device registers and the actual delays must be checked against controller documentation and the training result.

Reducing tRCD allows the read or write command to be issued earlier. If the bus is busy, the controller will still have to wait. When reading a new row, tRCD and CL occur in sequence: the row first becomes ready for RD, and then the delay before data output begins.

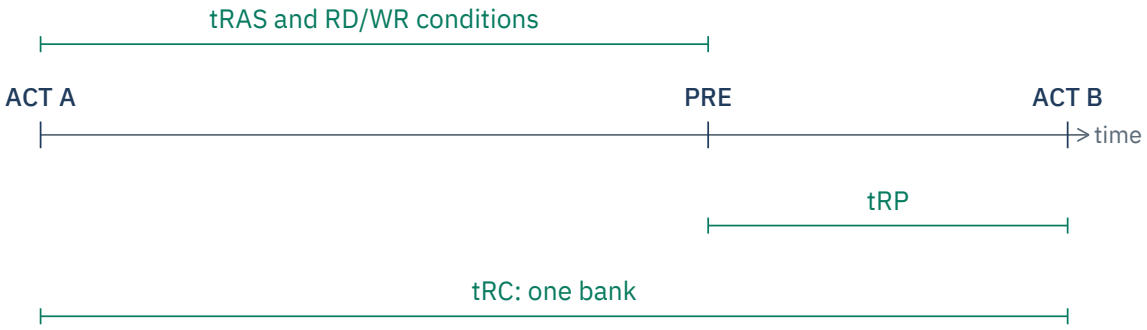
8.3. tRP: bank preparation for the next activation

tRP specifies PRE→ACT for the same bank. After a row is closed, the bitlines must return to their initial state so that the next weak signal can be detected correctly. Reducing tRP shortens this part of a row change, provided PRE has already been issued in compliance with all constraints. Figures 37–38, p. 152, show how READ, PRE, and the next ACT fit together.

tRP begins after PRE and does not include the wait for that command itself. During a row conflict, the controller may first wait for tRAS, tRTP, or write recovery, then issue PRE; only then does tRP begin. If the bank is already closed and ready, there is no need to wait for this interval again.

The SK hynix DDR5 datasheet distinguishes PREpb for one bank, PREsb for the same bank number in all groups, and PREab for all banks of the selected device. All of these descriptions use tRP for the interval. For the name tRPab in other memory families or BIOS menus, the corresponding constraint and its scope need to be established.

Figure 13. One bank cycle: ACT → PRE → ACT



Schematic scale. ACT→PRE: RAS; PRE→ACT: RP; ACT→ACT: RC. Only the commands and intervals of the row cycle are shown.

8.4. tRAS: minimum row active time

tRAS, row active time, specifies the minimum ACT→PRE interval for one bank. After activation, the row must remain active long enough to complete the necessary internal actions, including charge restoration. Data may become available for column access before the entire cycle is allowed to end.

The documentation specifies the maximum row active time and the refresh requirements. For PRE, the controller compares the tRAS boundary measured from ACT with the boundaries measured from the latest reads and writes. If tRTP or write recovery permits PRE later, reducing tRAS further will not bring closing forward.

In the sequence ACT, RD after tRCD, PRE after tRTP, the sum tRCD + tRTP sets one ACT→PRE boundary. The independent tRAS requirement applies at the same time. Thus, neither tRAS = tRCD + tCL nor tRAS = tRCD + tRTP describes a general rule: all the conditions in this sequence determine the permissible PRE time.

8.5. tRC: the full cycle of one bank

tRC is the minimum interval between successive ACT commands to the same bank. Before activating it again, the controller must wait until the previous row can be closed and then complete precharge. The specification therefore relates tRC to tRAS and tRP: their sum sets the lower bound for this sequence.

The minimum requirements use tRC(min) ≥ tRAS(min) + tRP(min). After conversion to cycles, nRC(min) ≥ nRAS(min) + nRP(min) is checked separately: the inequality must also hold after rounding. The actual interval between ACT commands can be longer because of reads, writes, refresh, or a busy interface. The BIOS may calculate tRC from other values and provide no separate field.

Shortening the bank cycle helps a workload that frequently changes rows in one bank, if this cycle is what delays the next ACT. It does not speed up reads from an already open row. tRRD and tFAW specify ACT intervals between different banks; the next chapter covers them.

> When modeling PRE, it is useful to write: PRE time ≥ max(ACT + tRAS, RD + tRTP, reference end of write + tWR). Only events in this bank and this mode apply. The next ACT must then satisfy tRP and its own tRC constraint.

09. Activations: RRD and the sliding FAW window

Different banks can be prepared in parallel. Activation frequency is limited by the intervals between ACT commands and the number of activations already issued within a sliding window.

9.1. Constraints on issuing ACT

An activation affects many cells and row sense amplifiers at once, placing a load on the device's internal resources and power supply. Even ready banks therefore cannot be activated in arbitrary numbers almost simultaneously.

tRRD limits the interval between ACT commands to different banks of the corresponding device/selected rank. DDR5 documentation uses tRRD_S for different bank groups and tRRD_L for different banks within one group. Another ACT to the same bank is allowed once tRC has been satisfied and precharge has completed.

The S/L distinction is related to shared internal bank resources. The datasheet specifies intervals for each transition scope.

> tRRD_S/L: ACT → ACT in another bank; S—a different BG, L—the same BG. Reducing these values affects the stream of row openings. Repeated reads from an already open row do not create these constraints anew.

9.2. Four activations in a sliding window

tFAW stands for Four Activate Window. The fifth activation counted by the rule is allowed after the window that started at the first of the preceding four has ended. Once the fifth is issued, the check shifts: the sixth is compared with the second.

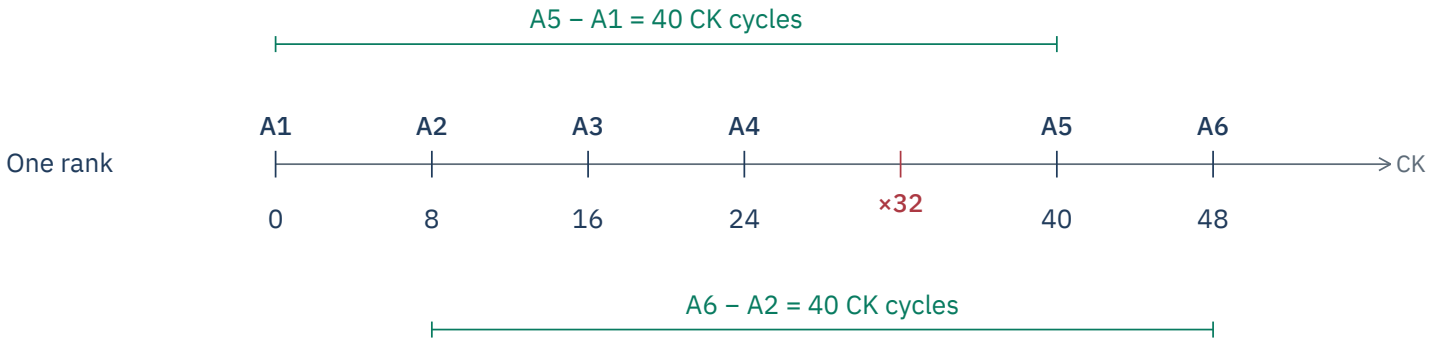
Let the ACT times be A1, A2, A3, and so on. For a sequence consisting only of ACT commands, A5 – A1 ≥ tFAW must hold, followed by A6 – A2 ≥ tFAW, and so forth. At the same time, tRRD, the readiness of each bank, tRC when a bank is reused, and command interface occupancy still apply.

Consider an illustrative sequence in one rank with a window of 40 CK cycles and ACT commands spaced 8 CK cycles apart: 0, 8, 16, 24. The fifth cannot be issued at 32: only 32 CK cycles have elapsed since the first. At 40, the first window has expired, and the fifth becomes permissible if the other conditions are met. The sixth is allowed no earlier than 48, since it is now compared with the ACT at 8.

In this example, 16 CK cycles separate the fourth and fifth ACT commands with a window of 40 CK cycles. FAW limits activation frequency; banks that are already open may remain active after the window shifts.

A(i+4) – A(i) ≥ tFAW
next ACT ≥ max(RRD boundary, FAW boundary, bank readiness, ...)

Figure 14. The fifth and sixth ACT in a sliding window



From A4(24) to A5(40), 16 CK cycles elapse with a 40-cycle window.

Example based on SK hynix Table 322: one rank, x16 with a 2 KiB page, 6400 MT/s. tFAW = 40, tRRD_S = 8 (CK cycles); BG0→1→2→3→0→1, different ready banks. REFsb/RFMsb are excluded.

9.3. Relationship between FAW and 4 × RRD

Consider five ACT commands to different ready banks of one selected rank, with no REFsb refresh commands or RFMsb additional maintenance commands between them. There are four intervals between the first and fifth ACT commands: first to second, second to third, third to fourth, and fourth to fifth. If each interval is at least a common lower bound, their sum is at least four times that bound. When tFAW does not exceed the sum of four such minimum intervals, the window is already satisfied and adds no wait before the fifth ACT.

The bound for each interval depends on bank placement: tRRD_S applies between different bank groups, while tRRD_L applies between different banks within one group. A mixed sequence therefore needs to account for both constraints. The equality tFAW = 4 × tRRD_S describes only the case in which four short intervals already cover the window; by itself, it does not specify a permissible tFAW for every device.

In Table 322 for DDR5-5600–6400 in the SK hynix 32Gb DDR5 SDRAM datasheet, Rev.1.2 (07.2025), on PDF page 473, the DDR5-6400 minimums are tRRD_S = 8 CK cycles and tRRD_L = max(8 CK cycles, 5 ns). At 6400 MT/s, one CK cycle is 0.3125 ns, so the tRRD_L constraint corresponds to 16 cycles. For x4/x8 devices with a 1 KiB row, tFAW = max(32 CK cycles, 10 ns), or 32 cycles. For x16 with a 2 KiB row, tFAW = max(40 CK cycles, 12.5 ns), or 40 cycles.

For x16, the sequence can access groups BG0, BG1, BG2, BG3, and then BG0 again, choosing a different ready bank each time. The first four ACT commands occur at 0, 8, 16, and 24 CK cycles. Under the short interval, the next ACT would be allowed at 32, and the return to BG0 would already satisfy the long interval of 16 cycles. But the tFAW window of 40 cycles delays the fifth ACT until 40—exactly as shown in the figure. For the x4/x8 devices under consideration, the window is 32 cycles, so with the same transitions it adds no delay to the four intervals of 8 cycles each.

9.4. Scope of the FAW window and additional events

The window applies within the corresponding device and selected rank. One shared FAW constraint applies across multiple banks of the selected set.

3DS (3D-stacked DRAM) is a device with several dies in one stack. Its command model distinguishes logical ranks, so the window's scope and some other intervals are specified separately; section 14.3 gives an example of this distinction.

In the DDR5 mode under consideration with a monolithic die, REFsb and RFMsb also count as activations for tFAW. The figure shows a sequence without refresh or the additional RFM maintenance associated with row activity (section 12.6). Additional logical-rank conditions apply to 3DS.

Reducing the effective FAW can make the activation stream denser if the window was delaying the next command. With accesses to open rows or stricter RRD and bus constraints, the schedule may remain unchanged.

10. Column access and CCD intervals

An open row allows repeated access to its columns. The rate of this stream depends on the intervals between commands and data bus occupancy as well as CL.

10.1. Data transferred by one access command

RD or WR selects a portion of the active row and starts a burst—a sequence of data transfers. Burst length, BL, specifies the number of transfers on each DQ line.

In the example, BL16 and 32 payload lines in the subchannel give 16 × 32 / 8 = 64 bytes. One x8 device transfers 16 bytes; four devices together transfer 64. The sixteen transfers of BL16 take eight CK cycles.

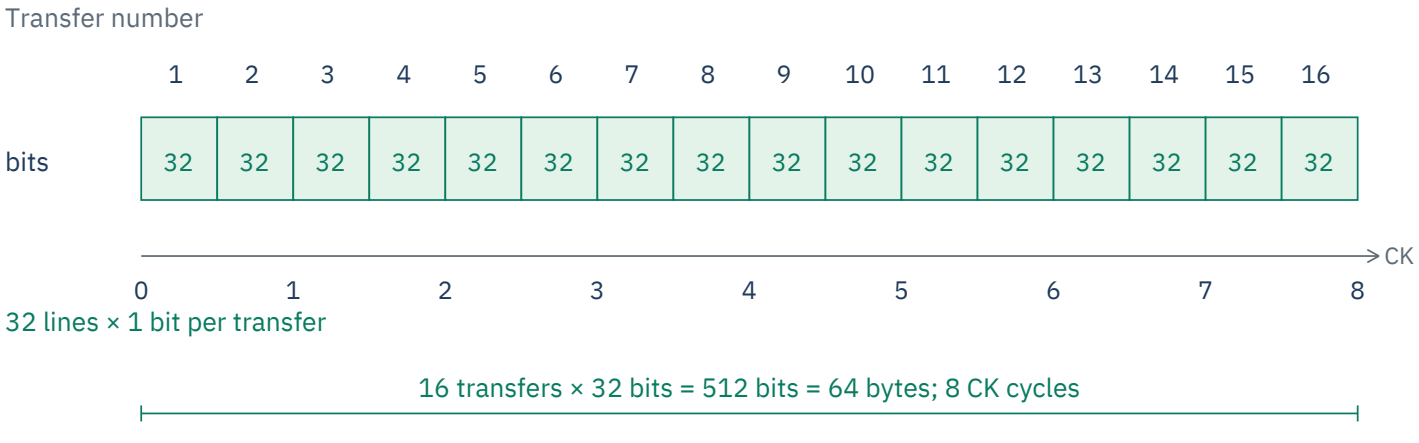
DDR5 uses 16n-prefetch: data is prepared internally in parallel and sent sequentially over the external interface. In the basic mode, there are 16 transfers on each line. Section A.4 explains how this differs from CPU prefetching.

In burst chop 8 mode, BC8, the external sequence is shortened to eight transfers. With 32 payload lines, this is 32 bytes and 4 CK cycles. On-the-fly selection, OTF, means that the command itself selects a supported option. The device datasheet must be checked for the availability and conditions of these modes.

BL as the actual number of transfers must be distinguished from the RBL/WBL values used in the documentation's calculations. For example, SK hynix Table 41 retains RBL = WBL = 16 for BC8 OTF, even though eight data units are transferred. With CRC, which adds check data to detect transfer errors, the values used in calculations may also change. Formulas must be applied with the mode-specific notes taken into account.

Burst size, bytes = BL × payload DQ width / 8
Burst duration = BL / 2 CK clock periods

Figure 15. BL16 over 32 payload lines



Illustrative example: one subchannel, 32 payload DQ lines; BL16 without CRC. One bit per line per transfer: 16 × 32 bits = 64 bytes. The scale is in CK cycles: the final transfer starts at 7.5, and the full slot ends at 8.

10.2. First-response latency and continuous data transfer

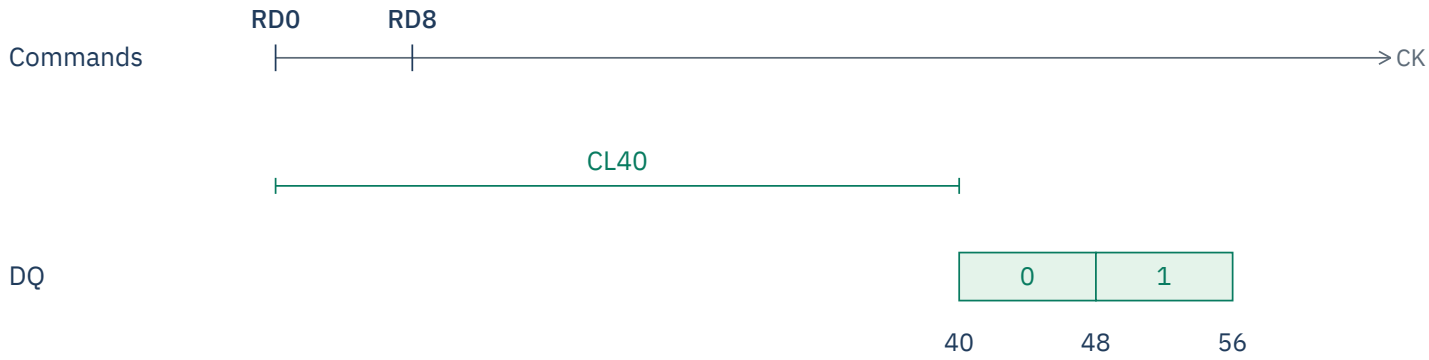
With two reads from open rows, the second RD command can be issued before data from the first request appears. This is a pipeline: operations overlap, but their data bursts must use the shared DQ bus without conflicts. The time to the first response and the interval between responses are therefore governed by different constraints.

tCCD is a family of constraints between column commands. In Table 324 of the SK hynix datasheet under consideration, for READ→READ, S means different bank groups, M means different banks within one group, and L means the same bank. These definitions apply to that table; a separate M field may be absent from the BIOS. WRITE→WRITE and WRITE→READ use their own variants and conditions.

In Table 322 of the selected SK hynix datasheet, M values match the corresponding L values. In the faster ranges covered by Tables 323–324, they differ, although the address-related meanings of S/M/L remain the same in these tables. Matching numbers in one range do not make M and L interchangeable.

A BL16 burst takes eight CK cycles. A longer interval between commands may be required because of internal constraints, bank placement, data direction, and the mode.

Figure 16. Latency and the spacing of a read stream



Illustrative example: one rank, different BGs; BL16 without CRC; CL = 40, RD→RD = 8 (CK cycles). Data 0: CK40–48; data 1: CK48–56. No direction changes or contention.

10.3. tCCD_L_WR and tCCD_L_WR2: write sequence requirements

A full write replaces the entire specified portion of data. A partial write must preserve some of the previous values. For read-modify-write, RMW, the device first obtains the old data, combines it with the new data, and writes the result. This is an internal operation that may require a different interval between external commands. For example, a write mask may allow some bytes to change while preserving others.

tCCD_L_WR and tCCD_L_WR2 distinguish write sequences. In SK hynix Table 45, for x8/x16, a second BC8 write or a partial BL16 write requires RMW and falls under WR, while a full BL16 write with no partial write falls under WR2. The choice depends on the mode and address scope. The device performs RMW internally, without a separate external RD command.

There is a physical reason for this distinction. In the device under consideration, on-die ECC combines 128 data bits and 8 check bits into one codeword—a unit that the protection code processes as a whole.

For x4, a BL16 burst transfers only 4 × 16 = 64 bits, so even a full external write replaces only half of the internal word and requires RMW. For x8, BL16 transfers 128 bits; for x16, two such codewords are processed in parallel. A “full burst” and a “full codeword” therefore describe different scales, and the WR/WR2 rules must be read with x4/x8/x16 and the write mask in mind.

The shorter WR2 interval applies when the conditions for the corresponding sequence are met. In the BIOS, these distinctions may appear as several fields or remain hidden in the controller’s rules.

The names tCCD_M_WR and tCCD_M_WTR describe separate cases in Tables 323 and 324 for this device. The suffix sg in a monitoring application is insufficient to map a field unambiguously to all these variants. A mapping must distinguish the same bank, different banks within one group, and different bank groups.

Family	Sequence	Main condition
tCCD_S / L / M	READ → READ	Bank placement and device conditions
tCCD_*_WR	WRITE → WRITE	Write path and internal conditions
tCCD_L_WR2	WRITE → WRITE, special case	Conditions for the short interval are met
tCCD_*_WTR	WRITE → READ	Write latency, burst, and post-write interval
tCCD_WTRA	WRITE → READ AP	Additional PRE eligibility condition

10.4. Bank group interleaving

If the access sequence allows different groups and their short intervals to be used, the controller can more easily fill the bus without gaps. This is one reason why the same frequency and CL do not guarantee the same bandwidth for two address streams.

If intervals between column commands leave gaps between payload bursts, reducing them can make the transfers denser. Once transfer is continuous, further reductions will not make the physical bus wider. Further increases in useful throughput require other options: more bits transferred per second, fewer direction changes, or a better distribution of requests.

10.5. Error correction, ECS, and CRC transfer checking

On-die ECC corrects a single-bit error within a DRAM codeword before the read data is output. Protecting the external DQ transfer requires controller-side mechanisms and an appropriate memory organization. The capabilities of system ECC depend on the platform and the protection scheme.

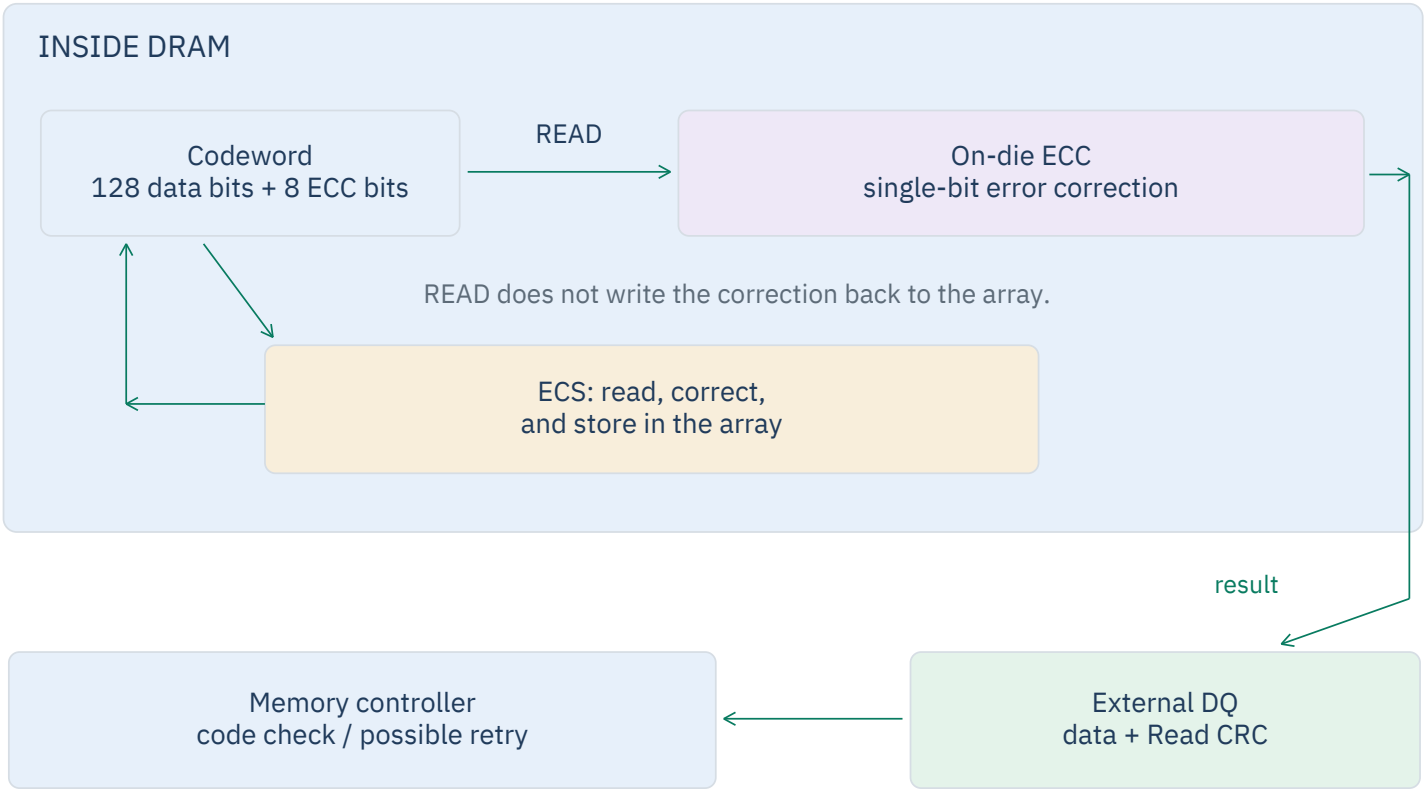
In the device under consideration, a normal READ outputs the corrected result. A normal READ does not write the correction back to the array. Error Check and Scrub, ECS, stores corrected data through an internal read, correction, and writeback. ECS counters record the results of this scan according to the filter settings. Corrections during normal reads are not included in that count. This distinction is specified in §4.36 of the SK hynix datasheet, p. 300, and in the ECS description on pp. 301–305.

CRC, a cyclic redundancy check, checks data transfers between the controller and DRAM. For WRITE, the controller generates the check code; for READ, the memory does. A mismatch allows an error to be detected. With Read CRC, the controller can repeat the transaction; with Write CRC, it is responsible for the specified retry and data consistency. Read checking and write checking are enabled separately. In the DDR5 device under consideration, Write CRC is incompatible with the data mask: these features cannot be treated as active at the same time.

Enabling CRC changes the schedule. In SK hynix Table 41, for BL16, the RBL or WBL used in calculations with the corresponding CRC check code is 18 rather than 16; the amount of payload data remains unchanged.

SK hynix Table 150 specifies a read CRC latency adder of 2 CK cycles at data rates above 6000 and up to 8800 MT/s, and zero in the 2933–6000 MT/s range. The controller accounts for this adder in the programmed CL through mode register MR0; it must not be added again to an already adjusted CL.

Figure 17. Internal correction and external transfer



CRC checks the transfer only when the mode is enabled.

The READ path in the device considered here. Data corrected during a read is not written back to the array; ECS stores the correction. When enabled, Read CRC is generated by DRAM and checked by the controller. Write CRC works in the opposite direction. System ECC is a separate layer of protection.

11. Writes, data direction changes, and PRE eligibility

The WRITE command, the arrival of data, a direction change, and cell recovery are separate events. Each has its own constraint.

11.1. A shared data bus and two signal sources

During a read, memory drives data onto DQ; during a write, the controller does. When the direction changes, one side releases the lines for the other. The receiver needs a strobe and data with the correct timing margins.

A mixed read/write stream requires pauses for direction changes—turnarounds. These pauses, the command-to-data delay, and the internal recovery of written cells have different reference points and purposes.

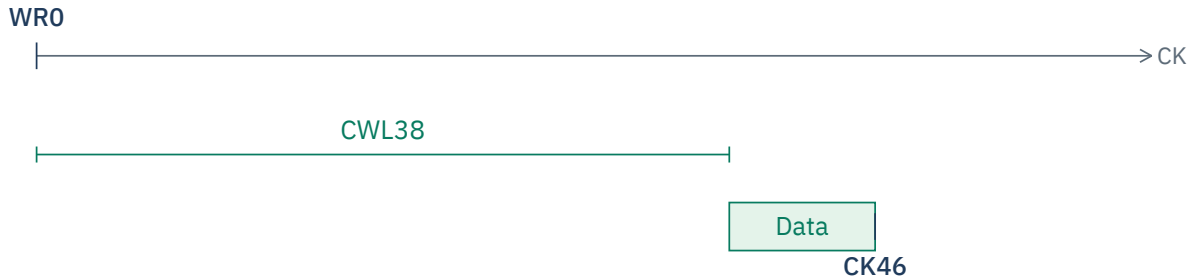
The discussion below assumes one selected rank, BL16, and consistent command reference points. A transition to another rank must additionally account for source switching, termination, and reception conditions.

11.2. CWL: interval from WRITE to the start of data transfer

CWL, CAS write latency, determines the delay before write data reception begins. In the verified datasheet, it is measured from the last cycle of WRITE to the first effective rising edge of DQS, excluding the preamble. In the main example, 16 data units are then transferred on each line. WBL in the formulas is the value used for calculations in the notes to SK hynix Table 41; section 10.1 covers the BC8 OTF and CRC exceptions.

In the DDR5 mode under consideration, CWL = CL − 2. With CL = 40, CWL is 38 CK cycles; the BL16 data transfer then takes another 8 CK cycles. The next action after the burst may require an additional interval.

Figure 18. WRITE and the arrival of data



DQS: first effective edge at CK38

Illustrative example of one write burst: BL16 without CRC; CWL = 38 CK cycles. WR at CK0; the first effective DQS edge is at CK38, the last transfer starts at CK45.5, and the first rising CK edge after the last data is at CK46.

Reducing CWL moves data reception closer to the command and also changes the calculated conditions for subsequent transitions. The controller and DRAM must use the mode consistently; a separate visible BIOS field does not guarantee that it can be selected independently of the other settings.

11.3. tWTR: interval from write data completion to READ

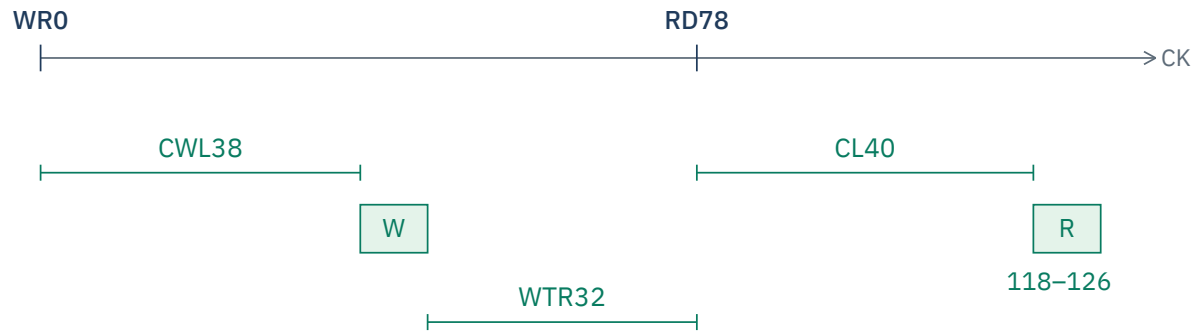
tWTR specifies the interval before a read is allowed after write data has been received. Starting from WR, CWL elapses first, then the write burst is transferred and the applicable interval is observed before READ. The reference boundary is the first rising edge of CK after the last write data in the adopted diagram. In a model measured in CK cycles, this corresponds to the burst completion boundary.

In SK hynix Tables 322, 323, and 324, the full interval between column commands is written as tCCD_*_WTR = CWL + WBL/2 + the corresponding WTR. The asterisk denotes the scope: the short path is between different BGs; the long path and, where defined, the intermediate path are within a BG.

In the illustrative example, CWL = 38 CK cycles, WBL = 16, and a post-write interval of 32 CK cycles give WR→RD of at least 78 CK cycles: 38 + 8 + 32 = 78. The Intel tWRRD field may represent the interval differently and apply adjustments, so its relationship to this formula requires confirmation.

Reducing WTR can shorten the wait for a read after a stream of writes. If the scheduler is still performing other writes or reading another resource, the reduction need not speed up the selected request. Large uniform streams may have few transitions, while mixed streams may have substantially more.

Figure 19. From write data to READ



First rising CK edge after the last write transfer: 46

Illustrative example: one rank; BL16 without CRC; CWL = 38, WTR = 32, CL = 40 (CK cycles). WR at CK0; write data at CK38–46; RD at CK78; read data at CK118–126. All other minimum intervals are satisfied.

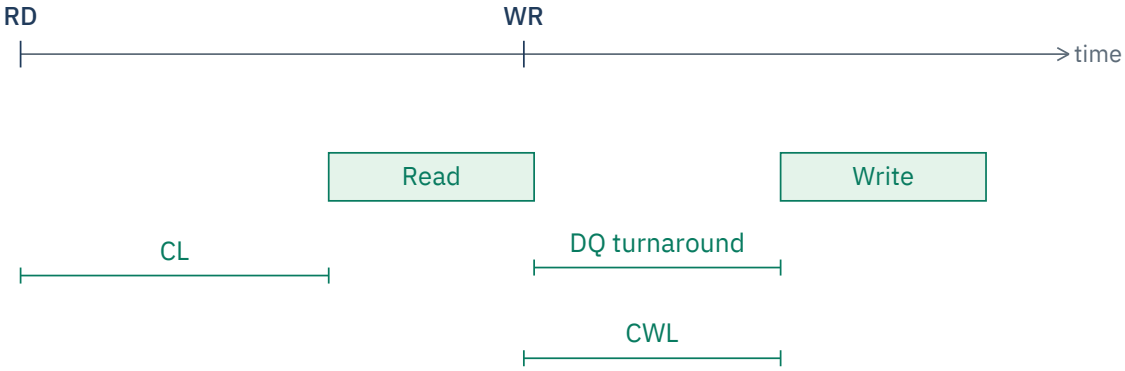
11.4. READ → WRITE: data direction change

After READ, memory outputs a burst, and the controller must then begin its own write burst. Command issue times must be chosen so that external data and DQS do not conflict. The calculation includes read latency, write latency, burst length, DQS offset, the preamble, the postamble, and rank conditions.

tRTW, tRDWR, and individual register fields may describe this transition at different levels. This does not imply a universal formula of the form CL – CWL + a constant for every DDR5 device. For a specific mode, the calculation must account for all the conditions listed above.

In the diagram, read data transfer ends, the handover interval specified by the protocol elapses, and write data transfer then begins. WR timing is chosen with CWL taken into account: the command can be issued before read data transfer has ended if the subsequent data transfer fits the permitted schedule.

Figure 20. Changing direction after READ



DQS conditions: offset, postamble, preamble

Schematic scale: data intervals are shown separately from commands. The gap depends on DQS, the preamble, the postamble, and the mode.

11.5. tWR and tRTP: intervals before issuing PRE

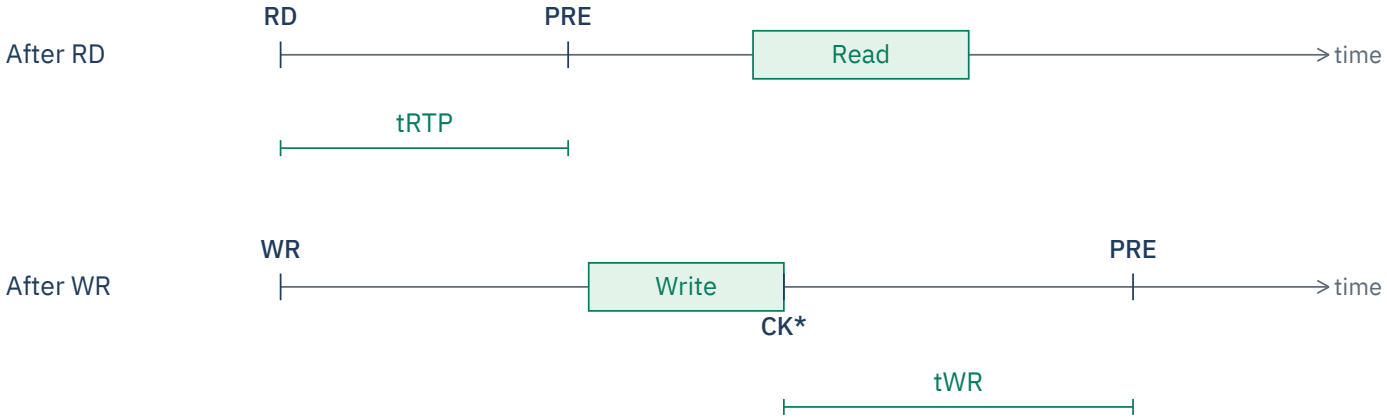
tWR, write recovery, is measured from the first rising edge of CK after the last write data to a permissible PRE in the same bank. The written information needs enough time to become reliably established in the array. This interval allows the array to prepare for closing after the external burst has ended.

The full WRITE→PRE path in a model measured in CK cycles includes CWL + WBL/2 + tWR. But PRE must also satisfy tRAS measured from ACT. If the minimum tRAS interval expires later, reducing tWR will not move PRE; if write recovery finishes later, it is what determines when the bank can close.

tRTP, read-to-precharge, specifies the external READ→PRE interval for the same bank without additionally adding CL and BL/2. In the pipelined path, data may already be traveling toward the output when the array is allowed to start precharge.

tWR and tRTP primarily affect the ability to finish working with the current row and move to another row or to maintenance that requires a precharged state. They do not automatically bring forward the start of an already scheduled read burst. The encodings of their values must be checked separately from the arithmetic minimum in nanoseconds.

Figure 21. Two different reference points before PRE



Top: tRTP is measured from RD; PRE can precede the external data. Bottom: tWR is measured from the first rising CK edge after the last write data. The scale is schematic; tRAS is checked separately.

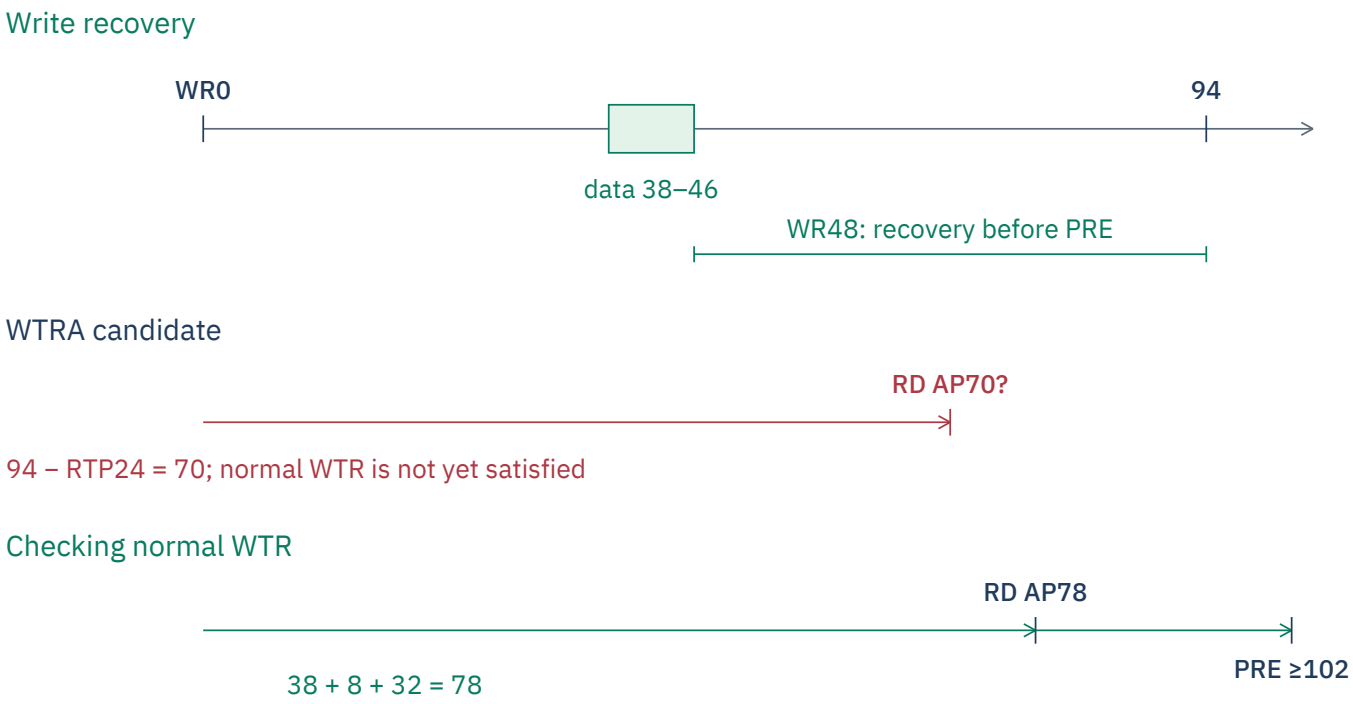
11.6. The tCCD_WTRA constraint

If WRITE is followed by READ with auto-precharge in the same bank, the future automatic PRE must wait for the previous write to recover. This introduces an additional lower bound, tCCD_WTRA ≥ CWL + WBL/2 + tWR − tRTP, under the conditions in SK hynix Table 324 and its note 2.

tRTP is subtracted because part of the wait before PRE will elapse after READ. The controller simultaneously checks the normal WRITE→READ constraint, tRAS, and the other rules of the mode.

For example, with WR at CK0, CWL = 38, WBL = 16, tWR = 48, and tRTP = 24, write recovery requires PRE no earlier than CK94. Under this boundary, READ AP is allowed no earlier than CK70: 38 + 8 + 48 − 24 = 70. With normal WTR = 32, however, the WRITE→READ interval is already 78 CK cycles. The later boundary, CK78, is selected; the tRTP boundary after it occurs at CK102. This example assumes that tRAS and the other conditions are met.

Figure 22. READ AP must pass both checks



Illustrative example: WRITE and READ AP in the same bank; BL16 without CRC, WBL = 16. Intervals in CK cycles: CWL = 38, WR = 48, RTP = 24, WTR = 32. WR at CK0; tRAS and all other conditions are satisfied. WTRA permits RD AP at CK70, but normal WTR postpones it to CK78. PRE is permitted no earlier than CK102.

12. Refresh and data retention

Refresh preserves data in the array. Its frequency, duration, and the set of occupied banks depend on the maintenance mode.

12.1. Periodic refresh of DRAM cells

Retention is the time for which a cell maintains a distinguishable state under specified conditions. Charge leakage limits this time, and it is not the same for all cells. Higher temperatures can increase leakage, so refresh requirements are tied to the device's temperature range.

ACT restores the selected row, but a program may leave other rows untouched for a long time. The controller must arrange refresh independently of the normal access stream. The DRAM's internal counter and logic select the corresponding rows; the processor does not send a separate address for every cell being refreshed.

Maintenance that is too infrequent reduces the data retention margin. A good signal on the external bus does not help retain charge inside the cells.

12.2. tREFI and tRFC: maximum average interval and minimum pause

tREFI specifies the maximum allowed average interval between refresh commands in the given mode—that is, the maintenance frequency. tRFC specifies the minimum pause after a command before the next permitted actions.

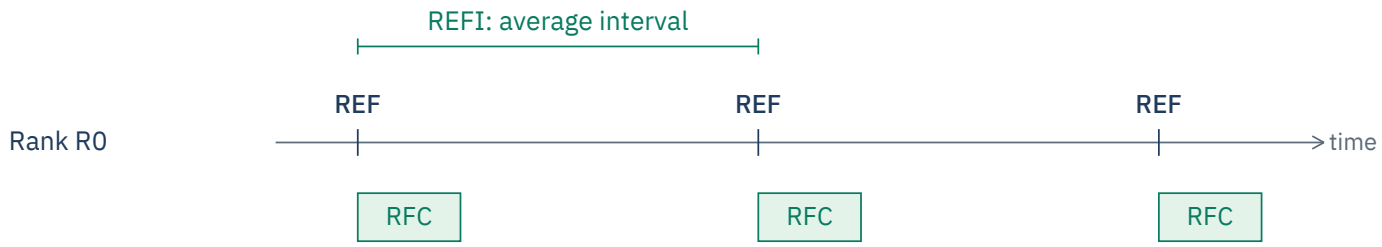
Before all-bank refresh, REFab, all affected banks of the device must be prepared: precharge must be complete and tRP must have elapsed. After REFab, they are unavailable for normal active accesses for the corresponding tRFC. The constraint applies within the selected device/rank.

Increasing the configured tREFI allows the controller to perform refresh less often. Reducing tRFC shortens the pause observed by the controller. The duration of internal refresh remains unchanged. With less frequent maintenance, the cells must retain charge longer; with a shorter pause, the internal operations must have enough time to finish.

In addition to the command interval and the duration of one operation, there is a full-array refresh window. For SK hynix 32Gb DDR5 SDRAM, Rev. 1.2, in Normal Refresh with REFab commands at 0 °C ≤ Tcase ≤ 85 °C, the base tREF window is 32 ms: 8192 commands are required within it. One REF may service several internal rows; the internal counter advances maintenance through the array.

The tabulated 3.9 µs is a rounded representation of tREFI, so the product 8192 × 3.9 µs does not have to equal 32 ms exactly. The tabulated constraint of 3.9 µs is retained for scheduling calculations. FGR and temperature-driven refresh rate increases follow their mode-specific rules. The window and intervals are specified in §4.13.4, Table 66, p. 190; internal maintenance is described in §4.13, p. 184.

Figure 23. Refresh interval and bank occupancy



Other ranks/subchannels may follow a different schedule.

Periodic REFab for the selected rank, on a schematic scale. tRFC is measured from REF; tREFI specifies an average interval and allows limited command postponement.

12.3. Normal refresh mode and FGR

In normal mode, tRFC1 applies to REFab. Fine Granularity Refresh, FGR, divides maintenance into shorter, more frequent operations: in the verified mode, REFab occurs more often, and tRFC2 applies to each operation. The average interval tREFI2 is half of tREFI1; tRFC2 does not have to be half of tRFC1.

In the SK hynix datasheet, for the 0–85 °C range, the average interval between REFab commands is 3.9 µs in normal mode and 1.95 µs in FGR. Above 85 and up to 95 °C, the intervals decrease to 1.95 and 0.975 µs. The temperature here is the IC's TCASE; readings from a DIMM sensor or SPD Hub cannot be equated to it without verification.

Doubling the refresh rate because of temperature and enabling FGR are independent causes: the same interval does not prove the same mode or tRFC.

For a 32 Gbit die, SK hynix Table 67 lists tRFC1 = 410 ns, tRFC2 = 220 ns, and tRFCsb = 190 ns. For 16 Gbit, the values differ: 295, 160, and 130 ns. This is the density of one die. A kit's total capacity of 32 or 64 GB does not determine its RFC: the device organization is needed. In the module capacity example in section 3.2, the same 32 GiB is obtained from devices of different densities; their refresh parameters must be selected according to the devices.

FGR mode is selected through a mode register. The device under consideration does not support on-the-fly switching.

Shorter pauses can occupy more time in total. For 32 Gbit at 0–85 °C, the fraction of time for which the selected set of banks is unavailable with evenly spaced REFab is 410 / 3900 ≈ 10.51% in normal mode and 220 / 1950 ≈ 11.28% in FGR. Each pause falls from 410 to 220 ns, while the total fraction occupied by these pauses increases slightly. The calculation does not account for overlapping work and does not show an application slowdown percentage.

Mode/command	Duration	Average interval, 0–85 °C	Scope
Normal mode, REFab	tRFC1	tREFI1 = 3.9 µs	All banks in the selected set
FGR REFab	tRFC2	tREFI2 = 1.95 µs	All banks in the selected set
FGR REFsb	tRFCsb	tREFI2 / n	One bank number across all BGs

n is the number of banks per group in the device under consideration.

12.4. REFsb: refreshing banks with the same number

REFsb selects the same bank number in each group: for bank No. 1, bank No. 1 in every BG of the selected device is serviced. In the DDR5 device under consideration, this command is available in FGR.

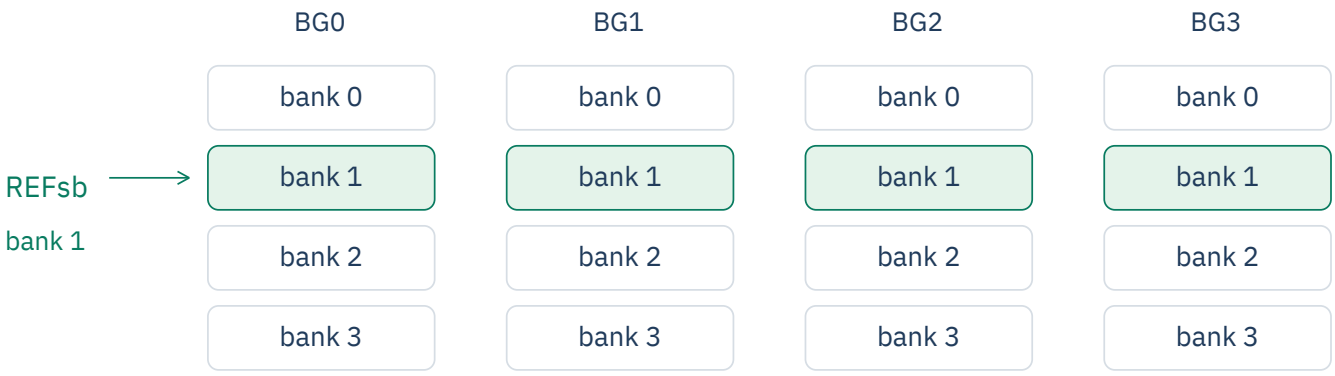
The other banks can service requests, but they are also subject to refresh constraints. In particular, tREFSBRD specifies the REFsb→ACT interval for other banks, and REFsb itself counts toward FAW for a device with a monolithic die.

The full set of bank numbers must be serviced to advance the internal counter. Repeating the same number before completing the set does not replace service of the others. With n = 4, the average interval between REFsb commands in the FGR example is 1.95 / 4 = 0.4875 µs. This interval is between individual REFsb commands; a full cycle services every bank number.

When REFab and REFsb are mixed, the complete set of bank numbers matters. If REFab is issued after REFsb for No. 0 and No. 1 without completing the set, those REFsb commands do not advance the global refresh counter in addition to REFab. The order of bank numbers within a complete set may vary. After REFsb, the serviced banks are closed, and the next row access requires ACT.

To map tRFCpb in the BIOS or a utility to DDR5 tRFCsb, the displayed hardware field and mode must be known. The name alone is insufficient.

Figure 24. REFsb: the same bank number in every group



A reduced schematic: four BGs with four banks each; bank 1 is selected. The other banks remain accessible subject to constraints, including tREFSBRD and the shared interface.

12.5. Refresh scheduling and postponement limits

The controller can move refresh to a more convenient time. The specification limits such postponement and early service: completed and deferred commands must be tracked, and the specified time windows must be observed.

In the verified document, normal mode permits up to four deferred REFab commands, while FGR permits up to eight. The corresponding maximum intervals between adjacent REFab commands are 5 × tREFI1 and 9 × tREFI2. Deferred commands remain a debt and must be performed while observing the minimum intervals. Self-refresh, SR, can be entered with a permissible debt, but its amount is retained after exit; further postponements must not exceed the total limit. In this mode, DRAM maintains the data itself; section 13.2 covers entry and exit. Debt retention is specified in §4.13.6 of the SK hynix datasheet, p. 192.

For example, if two normal-mode REFab commands have been deferred before entering SR, the debt of two commands remains after exit. Separately, SRX exit creates an additional refresh requirement, covered in section 13.2. The number of permitted postponements, the debt already accumulated, and the requirements during a mode transition must therefore be tracked separately.

The tRFC/tREFI calculation in section 12.3 estimates the fraction of time for which the selected resource is unavailable with evenly spaced REFab and no overlapping work. For REFsb, it is even less appropriate to treat this ratio as downtime for the entire channel: other banks and ranks can operate.

12.6. Temperature, RFM, and PRAC

A short test of cold memory does not cover all data retention conditions. Temperature changes during a sustained workload, and a different access pattern may expose errors in weak cells. One quick test does not establish that data is retained under these conditions when refresh overhead is reduced.

Refresh management, RFM, is additional maintenance related to row activity and device requirements. Its duration is specified by tRFM. RFM helps prevent data corruption from frequent activations: repeated row activations can disturb neighboring cells. Normal periodic refresh is still needed for charge retention. The RFMsb variant (Same Bank Refresh Management) provides time for this maintenance on banks with the same number—with one selected BA address—across all bank groups. In the device under consideration, this command is allowed only in FGR (§4.42, p. 327). RFM does not replace periodic REF; the internal algorithm of the specific die is not disclosed here.

With PRAC, Per Row Activation Counting, DRAM counts activations of individual rows. Through a back-off signal, it requires the controller to pause normal service and issue RFM. PRAC support must be confirmed for both the device and the controller.

Section 13.2 covers self-refresh and its exit conditions.

13. DRAM maintenance modes

Memory can reduce power consumption, retain data autonomously, and adjust its interface. Let's examine what happens during these transitions and when commands are allowed again afterward.

13.1. Power-down: entry, duration, and exit constraints

Power-down, PD, reduces memory power consumption. Entry is denoted PDE and exit PDX. Unlike self-refresh, this mode does not start autonomous refresh: the controller must exit in time and perform refresh. A maximum time in PD is therefore also specified. Internal operations already in progress may complete after entry.

Separate intervals apply between the last command and PDE and between PDX and the next command. tPD constrains time in the mode, while tXP specifies the time after PDX before permitted commands, with the exceptions listed in the documentation. tCPDED relates to the command path disable delay on entry. These parameters cover entering PD, remaining in it, and exiting.

The PDEN family specifies minimum intervals from commands to PDE. tACTPDEN applies to ACT, tPRPDEN to PRE, and tREFPDEN to REF. tRDPDEN accounts for read completion; tWRPDEN and tWRAPDEN account for writing and write recovery; tMRRPDEN and tMRWPDEN account for reading and changing a mode register.

In SK hynix Table 57, READ→PDE is specified as CL + RBL/2 + 1, and WRITE→PDE as CWL + WBL/2 + WR + 1 under the applicable conditions. These sums account for data arrival and transfer, and, for writes, recovery as well.

13.2. Self-refresh: autonomous data refresh

Self-refresh, SR, preserves data in the array through internal refresh and can operate without an external CK clock, but it requires the supply voltages to be maintained. Before SRE, all banks must be closed, tRP and the other intervals must have elapsed, and current bursts must be complete.

A DLL (delay-locked loop) adjusts delay to align signal timing with a reference clock. After a mode change, it may need time to return to its operating state.

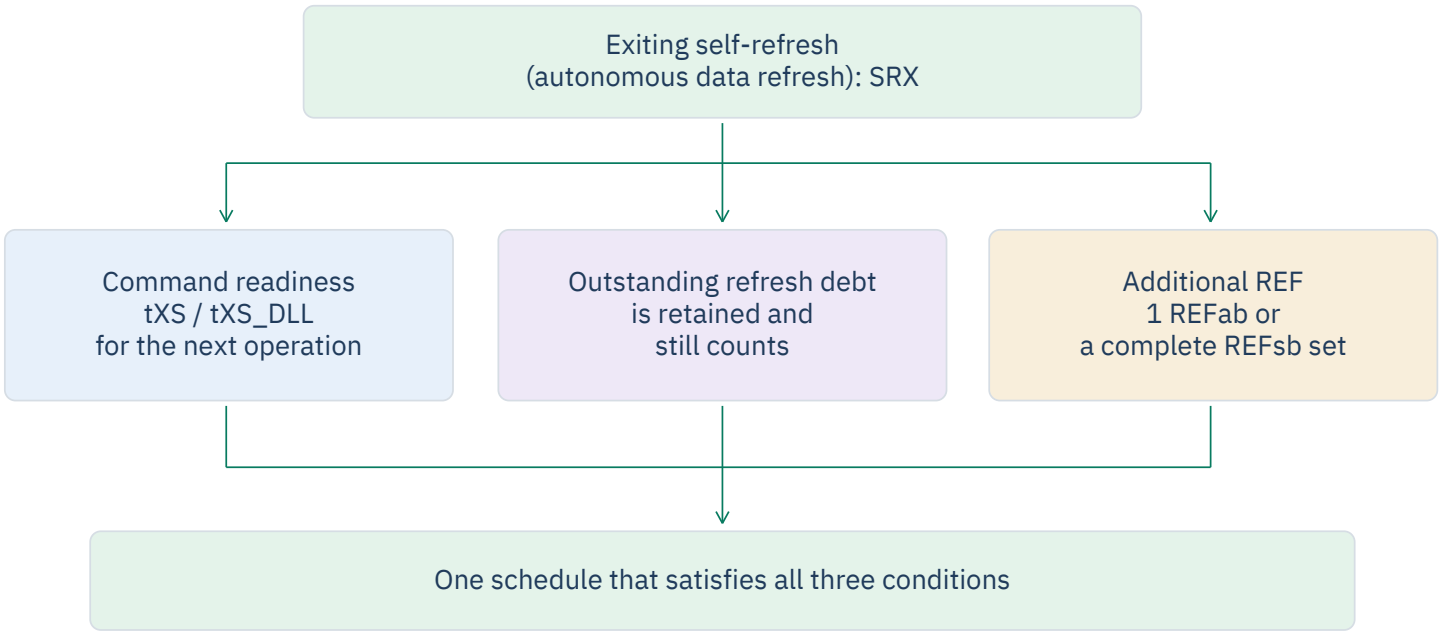
After SRX exit, tXS is observed before ACT, MRW, and other commands that do not require the DLL to be ready. RD, WR, and MRR require tXS_DLL. In the verified monolithic-die variant, tXS is tied to tRFC1, and tXS_DLL to tDLLK. These intervals are not automatically added together: the requirements for the specific next command are checked.

tCKSRX relates to CK clock stability before SRX exit. Similar BIOS names—tCKE, tCKCKEH, tXPDLL, or tSR—may be historical or platform-specific representations; without the register and documentation, their exact interpretation remains uncertain.

According to the verified datasheet, SRX must be followed by one additional refresh operation beyond the regular schedule: one REFab or a complete set of n REFsb commands, where n is the number of banks per group. This operation is not included in the average tREFI calculation and does not replace the next periodic refresh. If it is postponed beyond the first tREFI after exit, it counts toward the maximum number of deferred commands. The requirement is described in §4.9, p. 174, and the additional FGR conditions in §4.13.7, pp. 193–194, of the SK hynix datasheet.

In FGR, an incomplete REFsb set or an odd number of REFab commands before SRE may require additional maintenance under separate rules. Simply waiting for tXS does not replace this additional maintenance.

Figure 25. Three independent checks after SRX



After SRX, check the exit intervals, the outstanding refresh debt, and the additional refresh separately. An incomplete FGR set is subject to further rules.

13.3. Mode registers: MRR and MRW

Mode registers, MR, store device mode settings. MRR reads a register; MRW changes it. In SK hynix Table 17, both operations require closed banks; for 3DS, MRW has additional logical-rank conditions.

tMRR specifies the interval between these reads, tMRW the interval between writes, and tMRD the delay after programming before permitted actions. Changing a mode requires bringing the device into the necessary state and waiting for the setting to take effect. MRR and MRW access mode registers; RD and WR read and write program data.

13.4. ZQ calibration and the DQS oscillator

ZQ calibration adjusts the output circuitry and termination. Termination, including ODT inside the device, matches the line impedance and reduces signal reflections. ZQCal Start begins background calibration. After tZQCAL, ZQCal Latch accepts the result: DQ operations must have ended by this point, and the CA bus remains in Deselect (no command selected) during tZQLAT. During background calibration, the channel can continue transferring data subject to the mode's constraints.

The platform schedules repeated calibrations at an interval separate from these durations. The exact mapping of the BIOS fields tZQCSPeriod and tZQOPER to registers has not been established here.

The DQS oscillator tracks changes in the delay of the internal DQS path. The controller can use the measurement to maintain its settings. tOSCOM and tOSCOA constrain access to the result in manual and automatic modes. Which one corresponds to the BIOS field tOSCO cannot be established without documentation.

14. Parallelism and command scheduling

Banks can perform internal operations in parallel. The controller coordinates the timing of data transfers from these streams on the shared bus.

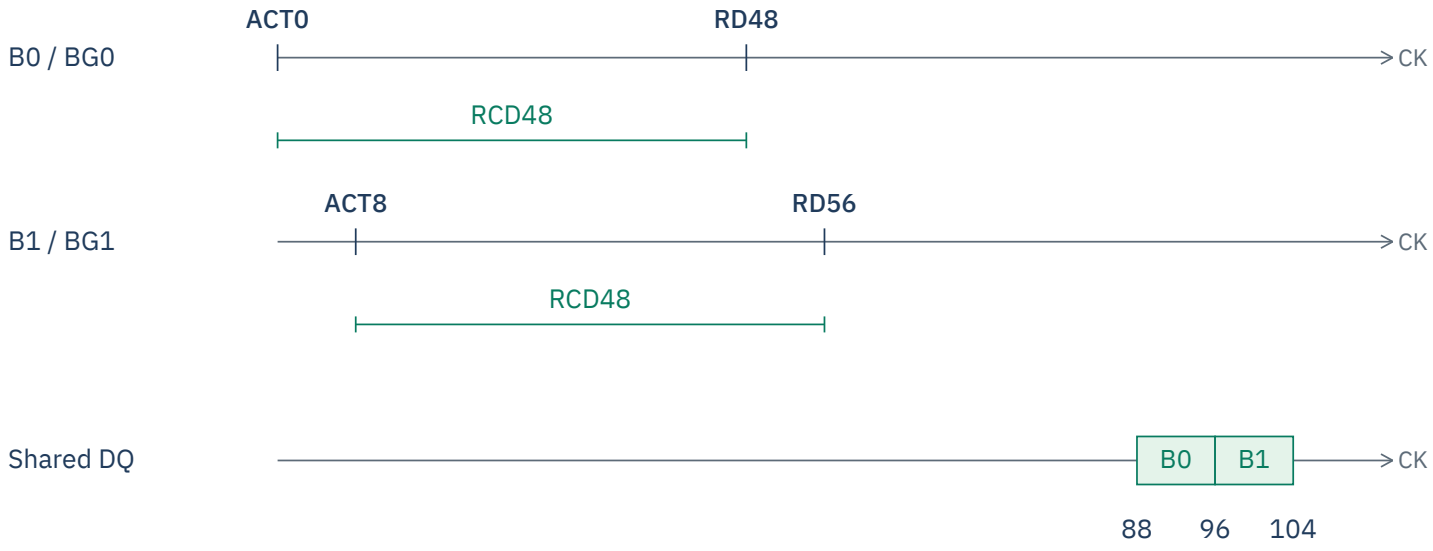
14.1. Two banks on one timeline

Consider two different banks in different BGs of the same rank. B0 starts activation at CK0. B1 can start activation at CK8 if tRRD, tFAW, and all other requirements are met. With tRCD = 48 for this example, the first RD is allowed at CK48 and the second at CK56. Preparation of the second bank starts 8 CK cycles after the first and overlaps it for 40 CK cycles.

Let CL = 40, BL16, and the interval between READ commands to the selected different BGs be 8 CK cycles. Then B0 data occupies CK88–96 and B1 data occupies CK96–104. The bursts use the bus in sequence, even though preparation of the banks partly overlapped.

Independent bank states let the controller overlap some internal delays with useful work in another bank. Bursts use the shared DQ line sequentially.

Figure 26. Overlapping the preparation of two banks



Illustrative example: B0/B1 in different BGs of one rank; BL16 without CRC; RCD = 48, CL = 40, RRD_S = 8, RD→RD = 8 (CK cycles). FAW is satisfied; the bursts use the shared DQ sequentially.

14.2. Interleaving, locality, and the request queue

Interleaving distributes accesses among banks, ranks, or subchannels so that resources can be prepared in advance for subsequent accesses. How much this helps depends on the address map and the request stream.

Row locality, the reuse of an open row, helps reduce the number of row changes. Switching too often between rows in the same bank adds PRE/ACT operations and disrupts a dense data stream. On the other hand, concentrating all requests in one bank can limit parallelism. The scheduler combines both opportunities within the available queue.

In a dependent chain of reads, the next element's address may be contained in the data from the previous element. The controller then learns that address only after the preceding read completes. A stream of independent requests, by contrast, lets it keep several operations in progress. Thus, two workloads with the same total read volume can use the same DIMM differently.

The controller can accumulate several WR commands and execute them consecutively, reducing the number of DQ direction changes. A read request to the same resource then waits for the series to end and for the WRITE→READ transition to become legal. This creates a tradeoff between lower direction-change overhead and lower latency for an individual read. Queue size, priorities, and series length depend on the controller.

14.3. Rank and subchannel parallelism

Another rank provides its own arrays and open rows. While one rank is busy with an internal operation, the controller can sometimes service another. But a rank change may require additional intervals, and the shared data lines still carry a coordinated stream.

A separate subchannel has its own command and data interface and allows more operations to run independently. The overall stream can be limited by IMC queues, CPU interconnects, or the number of parallel requests in the workload itself.

Refresh must fit into the same schedule. REFab blocks access to the selected set of banks for a specified interval; REFsb leaves some banks available, subject to restrictions. Performance depends on whether other useful work is available at that time and whether the controller can actually issue the commands for it.

In depth: 3DS logical ranks

3DS distinguishes SLR, the same logical rank within the stack, from DLR, a different logical rank. These designations do not replace dr, a transition to another rank in the Intel terminology discussed in section 16.1. In SK hynix Table 68, tRFC*_dlr constrains the transition to refresh of another logical rank and, under the stated conditions, is shorter than the full tRFC of the original rank. Expiration of the shorter interval does not mean that the first rank has finished refresh and is available for normal access. The next ACT in a different logical rank follows a separate rule.

15. Latency, bandwidth, and workload

Latency describes how long an individual request waits, while bandwidth describes how much data passes per second. The same setting can affect these metrics differently.

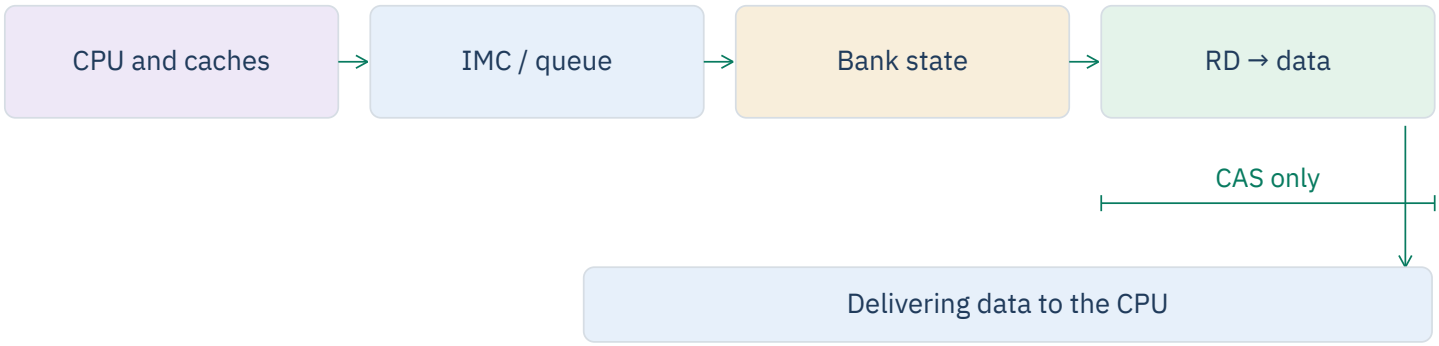
15.1. CAS latency and total request latency

In the formula CL × tCK, where tCK is the clock period, time is measured from a READ already issued to memory. Processing of the program's request began earlier: it included searching the caches, traversing internal interconnects, waiting in queues, and selecting a command. If a new row was needed, operations in the bank were added. After the data appears on the interface, it must be delivered to its consumer in the processor.

Even with the same CL and frequency, two reads can have different latencies. One hits an open row while the bus is free; another waits for a row change, the end of a write, or refresh. Both the average time and the latency distribution depend on the access stream.

The formula tRP + tRCD + CL describes only a simplified sequence for a row conflict, when PRE is already legal and all other resources are free. The sum of all primary timings does not describe every request. Moreover, some requirements apply simultaneously: a command must wait for the latest of the time boundaries calculated from the corresponding events.

Figure 27. What lies outside the CAS interval



The stages are logical, not drawn to a time scale. The path depends on caches, bank state, platform mode, and workload.

15.2. Interface bandwidth

The theoretical bandwidth of the interface equals the transfer rate multiplied by the useful data width. At 6000 MT/s and 32 bits, this gives 24 GB/s in decimal units. Two such subchannels provide a total of 48 GB/s if both continuously transfer useful bursts.

Command intervals, direction changes, row preparation, and refresh leave gaps on the bus. The program may also be unable to supply independent requests fast enough.

The operating time of bidirectional DQ is shared between reads and writes: the subchannel's maximum bandwidth cannot be obtained in each direction simultaneously. The program's useful data volume may also differ from interface traffic because of cache lines, prefetching, and additional transfers.

$$6000 \times 10^6 \text{ transfers/s} \times 32 \text{ bits} / 8 = 24 \times 10^9 \text{ bytes/s}$$

Suppose each next address becomes known only after the previous access completes. One access returns a 64-byte block, its full latency is hypothetically 60 ns, and accesses do not overlap. The stream of returned blocks is then limited by the following calculation.

64 bytes / 60 ns ≈ 1.07 GB/s

This counts all 64 bytes of the returned block; the portion useful to the algorithm may be smaller. Independent requests let waiting periods overlap and fill the interface more densely. Intel Memory Latency Checker uses dependent reads to measure latency; the 60 ns in our example is a hypothetical value.

15.3. Memory parameters on the critical path of a request

For dependent reads with frequent row changes, the stages of accessing a new row and the system path are especially significant. For a stream from open rows, column-command density and DQ utilization matter more. For mixed reads and writes, the distribution of operations by direction and direction changes matter. For frequent activations, RRD/FAW and bank readiness matter.

Reducing a timing helps as long as that timing is what delays the required action. Another parameter may then become the limiting factor. Even a substantial speedup in memory transfers sometimes makes little difference to the program's total execution time.

Under load, latency can increase because of queuing even while memory transfers a large amount of data per second. Maximum bandwidth and minimum latency are therefore usually measured in different scenarios. Intel MLC, for example, separately measures latency without load and at different levels of generated traffic.

Scenario	What to track	Other factors
Frequent row conflicts	PRE eligibility, RP, RCD, row cycle	Queue and shared DQ utilization
Stream of row hits	CCD, burst, shared DQ	Number of independent requests
Reads + writes	Direction changes and grouping of operations in the same direction	CWL, burst transfer, and WTR must be accounted for between WR and READ
Many ACT commands	RRD, FAW, ready banks	DRAM organization and device requirements
High queue occupancy	Parallelism, contention, refresh	Nature of requests and their distribution among resources

15.4. Test result interpretation

A test defines specific addresses, dependencies, thread counts, the share of reads and writes, prefetching behavior, and data volume relative to cache capacity. Its result applies to that workload and configuration.

A read test sees data after on-die ECC. Thus, the absence of mismatches does not prove that no single-bit errors were corrected inside the array; the number of visible errors also cannot be equated directly with the original number of cell errors. ECS counters, if the platform provides access to them, apply to a particular scan and its filters. A drop in bandwidth alone does not measure the number of ECC corrections: that conclusion requires separate observable evidence.

Theory lets us predict which resource should become available earlier: we need to trace the changed interval and the requests in which it participates. Practical testing will show whether this happened and whether data remained correct. The size of the speedup is measured for the chosen workload.

A higher frequency can shorten one CK cycle and increase the maximum data transfer rate, but the controller mode, encodings, and requirements in nanoseconds can affect the result. Comparing XMP profiles, which are sets of memory settings, therefore requires considering the entire verified operating mode.

16. Intel memory controller: tertiary timings, RTL, and refresh parameters

Controller fields define the DRAM schedule and interface configuration. Their relationship to standard timings depends on the platform.

16.1. RDRD, WRWR, RDWR, and WRRD notation

Intel's tRDRD, tWRWR, tRDWR, and tWRRD families denote intervals for four column-command sequences: read→read, write→write, read→write, and write→read. The suffix specifies the address scope. These are Intel controller fields, so their definitions are platform-specific. The discussion below uses TC_RDRD, TC_WRRD, and TC_RDWR from document ID 764981, Rev. 1.3; TC_WRWR comes from ID 767624, Rev. 1.3.

sg means same group, the same bank group; dg means different group. In the Intel definition considered here, dr means another rank on the same DIMM, and dd means another DIMM in the channel. A dr transition changes ranks; the ICs within one rank work together across the data bits.

If one DIMM is installed in the relevant channel, there is no transition to another DIMM. But a dr transition can still be used between the two ranks of that single module. Conversely, two 1R modules in one channel create a dd transition even though each module is single-rank. The specific subchannel wiring and register terminology must match the platform.

In the Intel DDR5 registers considered here, these intervals are measured in units of tCK, the CK clock period, while RTL is measured in QCLK controller cycles. The sg/dg scopes can be compared with the scopes of DRAM timing constraints. The numerical values may differ from tCCD_L/S or tWTR_L/S: the controller may account for additional requirements and represent the interval differently.

Direction	Relationship to DRAM theory	What to check
tRDRD_*	Interval between READ commands	BG/bank, burst, mode
tWRWR_*	Interval between WRITE commands	WR/WR2, partial write, scope
tRDWR_*	READ → WRITE	Data, DQS, direction change
tWRRD_*	WRITE → READ	CWL, burst, interval after a write
*_dr / *_dd	Rank / DIMM change	Data source, ODT, topology

16.2. RTL, QCLK, and memory training

Round-trip latency, RTL, in the Intel register considered here measures the time from READ until the first data returns to the controller. It includes propagation through the controller interface and the board, which CL does not account for. Prior queues and cache levels are not included in RTL.

The SC_ROUNDTRIP_LATENCY field is measured in QCLK controller cycles. With the Gear and frequencies verified, tQCLK = Gear × tCK. For example, a value of 30 QCLK cycles in Gear 2 at 6000 MT/s gives 30 × 2 × 2000 / 6000 = 20 ns. This converts the field value for the stated conditions. The definitions of SC_ROUNDTRIP_LATENCY, E020, and MC_BIOS_REQ, 5E00, appear in Intel Datasheet Vol. 2, ID 767624, Rev. 1.3.

A utility may convert the number or add an offset. Before applying the formula to its RTL reading, we need to confirm that it shows the field value in QCLK cycles without conversion or an offset. The usual tCL/tRCD/tRP values are already in CK cycles and do not need to be multiplied by Gear.

IOL, the name of an I/O path latency setting, RTL Offset, and training parameters align and configure the path at the platform level. The public RTL definition does not disclose their full mechanics. Training selects working timing positions and margins within DRAM requirements; a calculated equality alone is not enough.

For an unconverted RTL value in QCLK: time, ns = RTL × Gear × 2000 / MT/s

16.3. Controller refresh parameters

In the documentation for Intel register TC_RFP, E438 (ID 764981, Rev. 1.3), the clock for these controller intervals is designated DCLK. Its documented frequency is required to convert the intervals to physical time; DCLK, CK, and QCLK are not assumed to be equal based on their names.

tREFIx9 in Intel TC_RFP sets a maximum interval under the controller's rules and is measured in units of 1024 DCLK. Permissible refresh postponement depends on the mode, normal or FGR, and the balance of commands performed early and commands postponed; its limit cannot be inferred from the field name.

OREF_RI in the Intel register considered here sets, in DCLK cycles, the rank idle time after which opportunistic refresh becomes possible: refresh that the controller schedules during the rank's idle time. The field controls scheduling. A zero or any other value does not mean that mandatory array refresh is disabled.

For tCA2CS, tMOD, tWRPRE, and some other BIOS names, the exact mapping to registers of the current platform has not been established. The command interface, mode registers, write recovery, and PRE explain the related processes, but calculating a specific field requires board and controller documentation.

17. Summary model of memory operation and practical application

A BIOS parameter becomes easier to understand when we can see the command, resource, and specific waiting interval behind it.

17.1. A unified model of the operation sequence

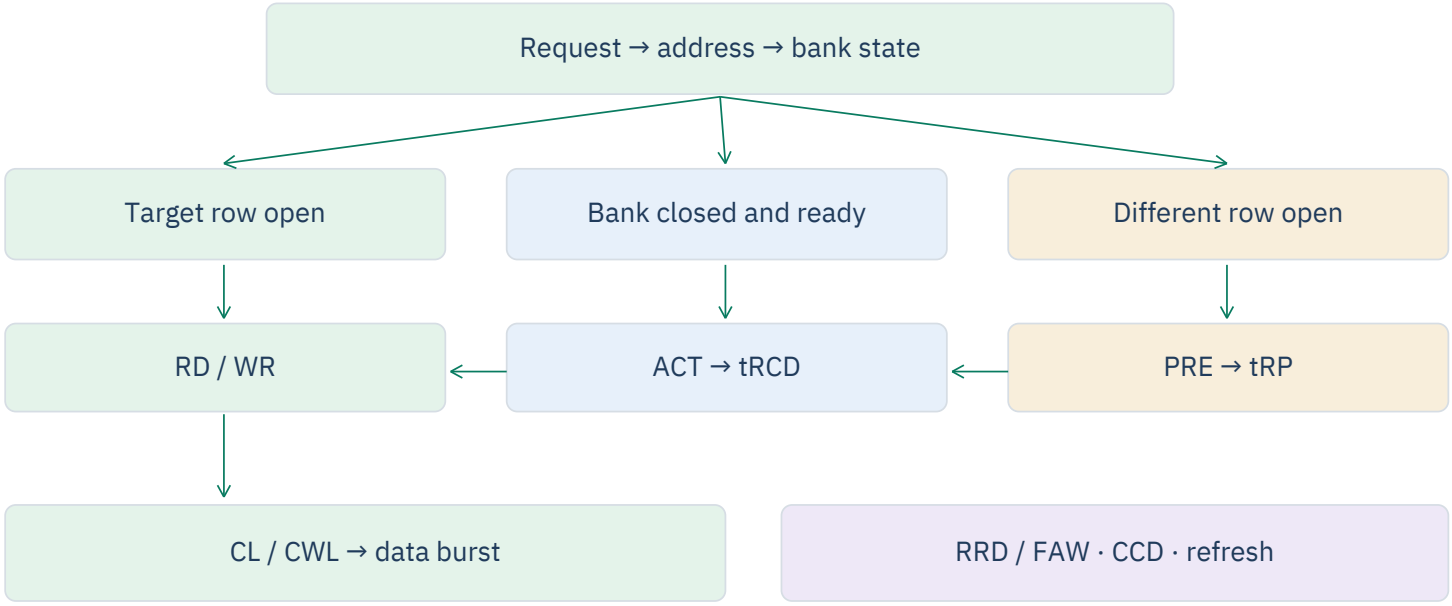
A DRAM request first reaches the memory controller. The address map determines the resources, and the scheduler checks the bank state and recent command history. If the required row is active, a read or write command can be prepared. If the bank is closed, ACT is needed. If another row is open, a legal PRE and preparation for a new cycle are needed first.

For a new row, tRCD determines when column access is allowed. Closing the row requires accounting for both tRAS and the conditions imposed by the most recent reads and writes; tRP applies after PRE, and tRC must be met before another ACT to the same bank. Activations of different banks are subject to RRD and the sliding FAW window.

RD and WR are scheduled with CCD, direction, rank, and burst transfer in mind. CL sets the time until read data begins to be output; CWL sets the time until write data begins to be received. WR→RD requires accounting for completion of the write and the interval after the write data; PRE after WR requires write recovery. Other banks can be preparing for subsequent operations while the shared DQ carries the current burst.

The controller includes refresh and transitions between device modes in the same schedule.

Figure 28. An overall map of commands and constraints



PRE ≥ max(ACT + RAS, RD + RTP, CK* + WR)

CK*: first rising CK edge after the last write data; WR→RD: CWL + burst + WTR.

Logical relationships, with no time scale. Write recovery and WTR begin after the write data. The controller incorporates refresh into the request schedule. The table below identifies where each group of constraints is explained.

Constraint group	Where it is discussed
Sequential stages and independent boundaries	chapter 7
Row: CL, RCD, RP, RAS, RC	chapter 8
Activations of different banks: RRD, FAW	chapter 9
Column commands and burst: CCD, BL	chapter 10
Writes and direction changes: CWL, WTR, WR, RTP, WTRA	chapter 11
Refresh: REFI, RFC, REFsb, debt	chapter 12
Modes and maintenance operations: PD, SR, MRR/MRW, ZQ	chapter 13
Parallel operation of banks, ranks, and subchannels	chapter 14
Request latency and data stream	chapter 15
Intel controller fields and units	chapter 16

17.2. Five criteria for timing analysis

From which event is the interval measured? Which next action is constrained? Which bank, group, rank, or bus does it concern? In which mode and units is the value specified? Which other requirement may remain more restrictive after a change?

A calculation using a platform parameter requires a publicly documented reference event, scope, and unit. Until this information is available, the field value cannot be used in the formula.

I recommend returning to the timeline whenever a formula starts to seem mysterious. Mark the command and the beginning and end of the data transfer, then mark the boundaries at which the next action becomes legal. This kind of diagram quickly shows where we need a sum, a maximum, a change of scope, or a window covering several preceding events.

17.3. Further reading

The practical guide “DDR5 Memory Tuning and Optimization on Intel LGA1700 / LGA1851 Platforms” continues this theory with tuning methods, voltages, training, and stability testing. When using it, relate each change to the command, resource, and constraint it affects.

Appendix A. DDR3 → DDR4 → DDR5: evolution of memory architecture

DDR3, DDR4, and DDR5 differ in transfer rate and memory architecture. Bank organization, internal prefetch size, and the distribution of requests among interfaces changed. These changes determine how many accesses memory can service at once and how the data bus is filled.

A.1. Shared principles and differences across DDR generations

DDR3, DDR4, and DDR5 retain the familiar sequence: open a row, access its data, close it if necessary, and move to another. Cells lose charge, so their contents must be restored. The discussion of ACT, READ, WRITE, PRE, and refresh in chapters 2 and 6 remains useful for all three generations.

The organization of this work changes. In the comparison table, the number of banks refers to one monolithic DRAM device, and the useful bus width refers to a standard UDIMM without system ECC. The relationship among devices, a rank, and a subchannel is discussed in chapter 3.

The comparison uses Micron 2-Gbit DDR3, Micron 8-Gbit DDR4, and SK hynix 32-Gbit DDR5. Bank count also depends on density: for example, the 8-Gbit DDR5 in SK hynix Table 1 has half as many banks as the 32-Gbit devices considered here. For x4/x8, that is 8 BGs with 2 banks each; for x16, 4 BGs with 2 banks each.

Mechanism / scope	DDR3	DDR4	DDR5
DRAM internal prefetch	8n (8 bits per DQ line)	8n (8 bits per DQ line)	16n (16 bits per DQ line)
Banks in one x4/x8 DRAM device in the selected datasheets	2 Gbit: 8 banks, no BGs	8 Gbit: 4 BGs × 4 = 16 banks	32 Gbit: 8 BGs × 4 = 32 banks
Banks in one x16 DRAM device in the same datasheets	2 Gbit: 8 banks, no BGs	8 Gbit: 2 BGs × 4 = 8 banks	32 Gbit: 4 BGs × 4 = 16 banks
Useful UDIMM interface without system ECC	One 64-DQ path	One 64-DQ path	Two 32-DQ subchannels
Main burst modes	BL8; fixed / on-the-fly BC4	BL8; fixed / on-the-fly BC4	BL16; on-the-fly BC8

BC means burst chop, a shortened transfer. In fixed BC4 mode, the length is set in advance; with on-the-fly selection, the command selects a full or shortened transfer in the permitted mode. In the SK hynix DDR5 datasheet, BL32 is available only for x4; the general example in this appendix uses x8.

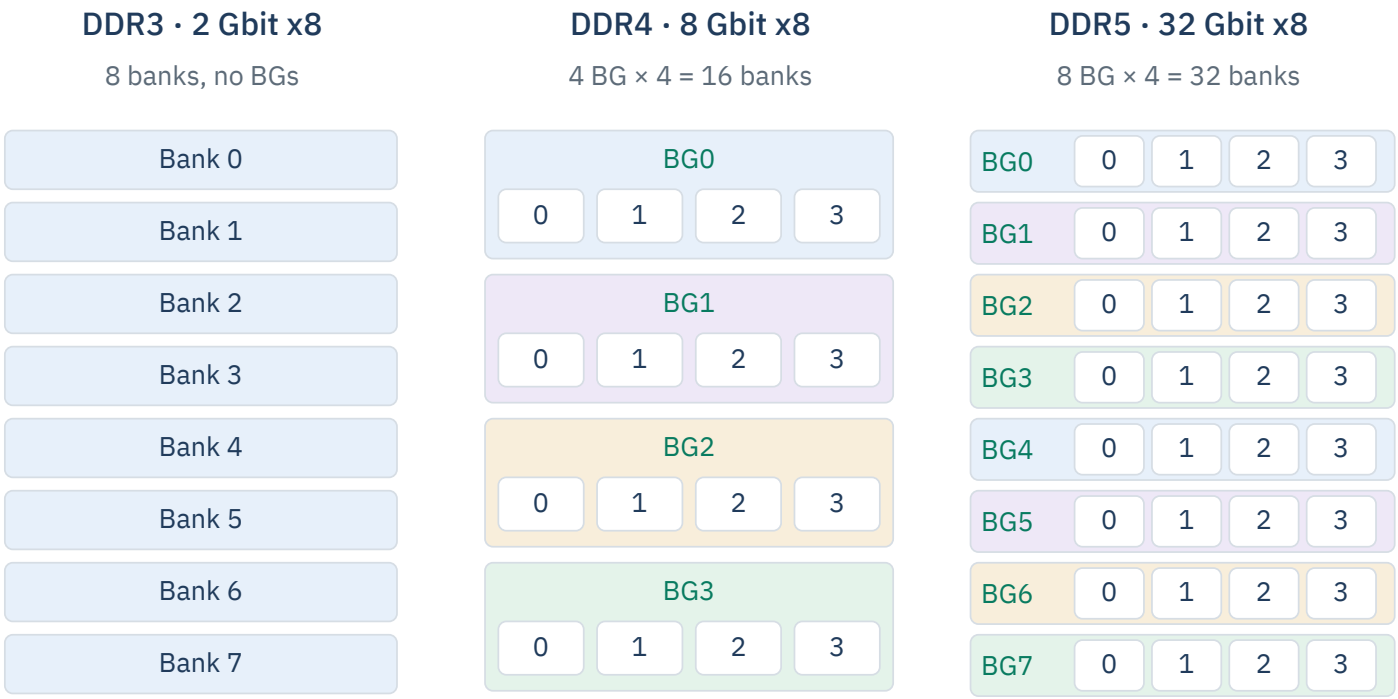
A.2. DDR3 → DDR4: architectural changes with 8n prefetch retained

Multiple banks allow the next access to be prepared while another is still in progress. Data from the selected banks exits through the device's shared interface. DDR4 organizes banks into bank groups, BGs: both the bank's readiness and its group matter for the next command.

The selected DDR3 devices have eight banks without BGs. DDR4 x4/x8 has four groups of four banks, twice as many banks; DDR4 x16 has two groups and the same eight banks.

In the applicable DDR4 constraints, S and L distinguish transitions between different BGs from those within the same BG. For example, tCCD relates two column read commands, and tRRD relates two ACT commands to different banks. Reads within one group are checked against tCCD_L; reads in different groups, against tCCD_S. For ACT, the tRRD_L/S scope is accounted for similarly. DQ utilization, bank states, and shared activation constraints still apply alongside these minima.

Figure 29. x8 banks: three specific organizations



Bank counts are for one DRAM chip.

Based on the Micron 2 Gbit DDR3, Micron 8 Gbit DDR4, and SK hynix 32 Gbit DDR5 datasheets. Each column represents one monolithic x8 device; numbers within a group identify banks. All BGs in the device use its shared DQ interface.

Alternating BGs allows a denser command stream: the controller selects ready requests in another group while the longer interval still applies to the previous group. Suitable requests must be present in the queue for this to work. Additional banks cannot parallelize a dependent chain of reads.

DDR4 nevertheless retained 8n-prefetch, internally fetching eight bits per DQ line. The new possibilities come from bank organization and access scheduling. A longer burst has a downside: as Micron explains, using it to transfer small amounts of data can be inefficient.

A.3. DDR4 → DDR5: transition to two subchannels

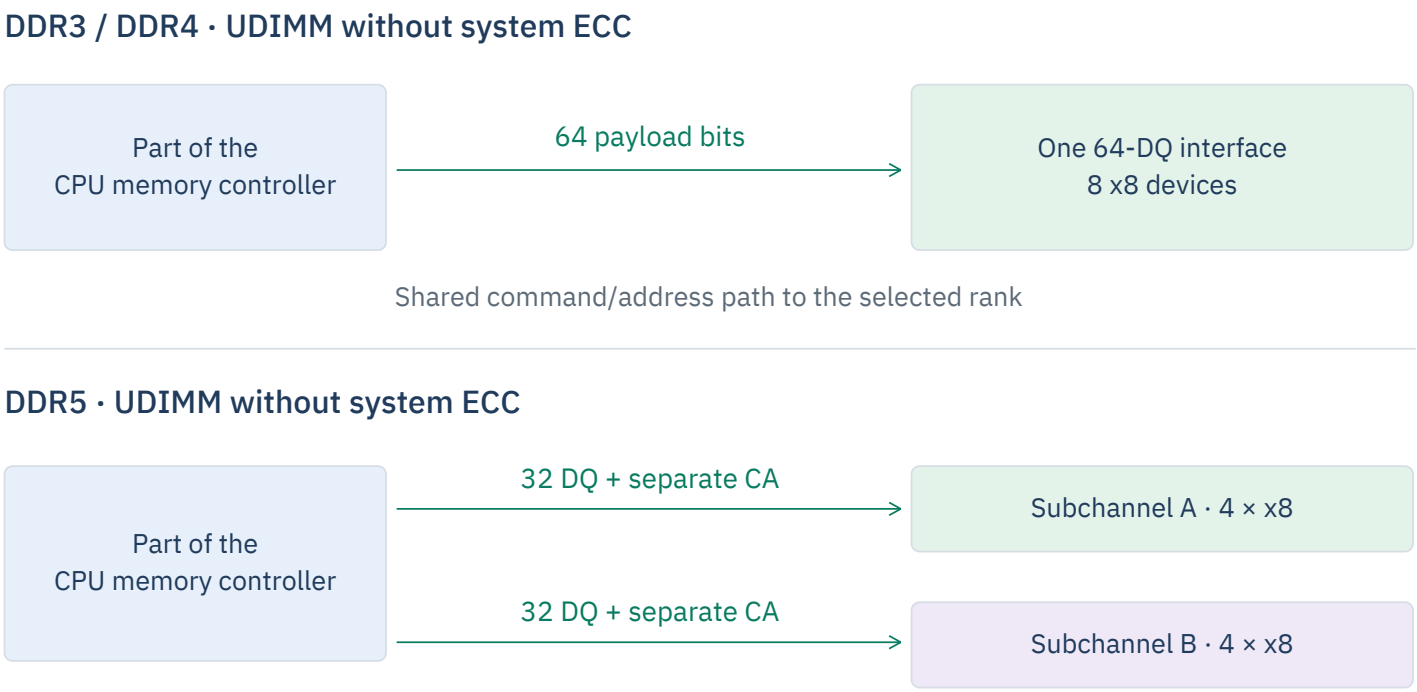
On a standard DDR3/DDR4 UDIMM without system ECC, data from the selected rank travels over 64 useful data lines. In DDR5, this type of module is divided into two subchannels with 32 useful data lines each. Each has its own command and address interface and its own data path. The controller can schedule accesses to them separately.

In the example with x8 devices, eight ICs work together for one transfer over 64 lines. Each 32-bit DDR5 subchannel has four. This explains the transfer width; module capacity also depends on device density and rank count.

The second subchannel provides another access path: one command sequence does not have to repeat the other. With suitable requests available, a transfer in one subchannel can overlap preparation or a transfer in the other. But the scheduler, CPU resources, power supply, and some platform constraints remain shared.

Two subchannels on a DIMM and two controller channels are different levels of organization. The number of available useful data lines and how they are used depend on the CPU, board, and installed modules.

Figure 30. 64 payload lines and two 32-bit subchannels



Regular UDIMMs without system ECC, using x8 devices and one selected rank per interface. Lines are grouped. DDR5 has two command/address and data paths connected to the same IMC.

A.4. BL16 transfer of 64 bytes over 32 data lines

A narrower path can transfer the same block in more transfers. For a full DDR3/DDR4 burst, take 64 useful data lines and BL8: eight transfers of eight bytes give 64 bytes. For one DDR5 subchannel, take 32 lines and BL16: sixteen transfers of four bytes give the same 64 bytes.

BL, burst length, denotes the number of transfers on each DQ line. The byte count and duration in full CK clock periods are calculated separately. Two transfers occur in one CK cycle. Thus, BL8 occupies four CK cycles and BL16 occupies eight CK cycles of the external burst, without additional modes. The byte count can be obtained only after accounting for the width of the useful data interface.

Internal prefetch describes the other side of this relationship: how much data DRAM prepares internally for sequential output. In the basic DDR3/DDR4 architecture, this is 8 bits per external DQ line; in DDR5, 16. For a full basic burst, this internal amount matches BL8 or BL16. Shortened BC4/BC8 bursts change the length of the external transfer while retaining the internal prefetch architecture. In the notation 4n/8n-prefetch, the numbers mean four or eight bits fetched internally per DQ line.

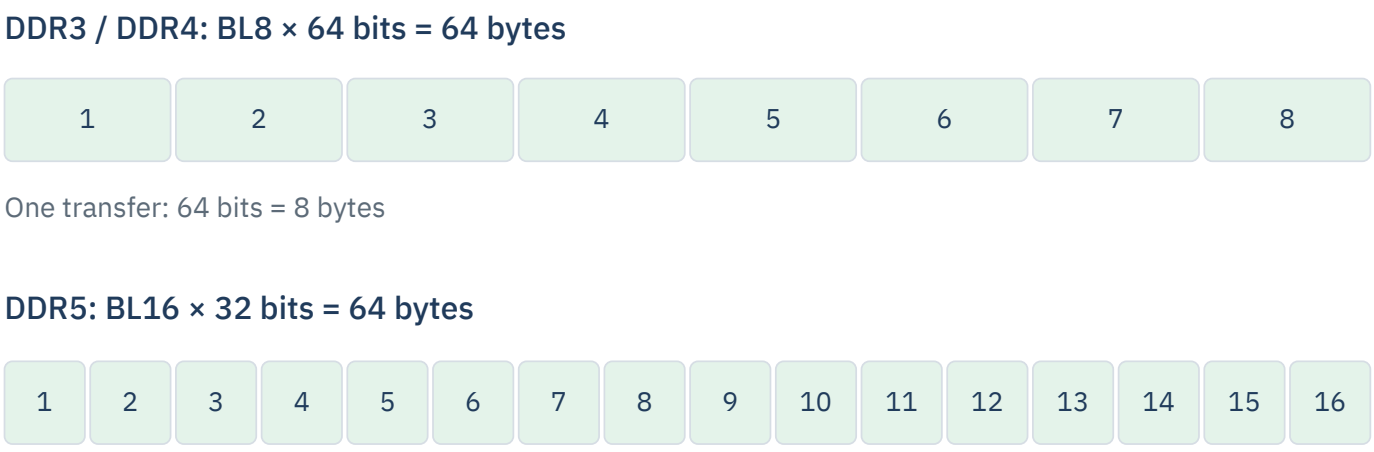
CPU prefetching works differently: the processor requests addresses in advance that the program may need. DRAM 8n/16n-prefetch describes the internal preparation of data for output over DQ.

Half the previous width with twice the burst length preserves the 64-byte amount, while the second subchannel can service another access. How fully both subchannels are used depends on the request stream.

DDR3 / DDR4: 64 bits × BL8 / 8 = 64 bytes
DDR5, one subchannel: 32 bits × BL16 / 8 = 64 bytes

> The comparison uses full bursts without additional ECC bits or CRC, the transfer check code. 64 bytes is the block chosen for this example. Internal row geometry and burst-transfer details are discussed in section 3.4 and section 10.1; the row is considerably larger than this amount.

Figure 31. The same 64 bytes with different transfer counts



Illustrative comparison of full bursts: DDR3/DDR4—64 DQ, BL8; DDR5—one 32-DQ subchannel, BL16. No additional ECC bits or CRC. The bar widths represent volume—the same 64 bytes. Transfer duration depends on burst length and the CK period. The cells show transfer numbers.

A.5. Timing names and differences in their conditions of use

The basic constraints are familiar across all three generations: a row needs time to prepare, a bank needs time to close, and the interface needs time to transfer data. Their exact values and reference events depend on the protocol and mode.

The constraint table describes normal access to the selected bank. It assumes additive latency AL = 0 and full BL8 bursts for DDR3/DDR4, and BL16 for DDR5. Additional latencies and CRC are excluded. The command time is its reference point on the manufacturer's diagram. In DDR5, READ occupies two command cycles, and CL is measured from the second.

Constraint	Initial → final event	Scope and meaning of the comparison
CL	READ reference point → nominal start of data output	Selected read. With AL = 0, this is the CAS part of read latency, not the entire CPU request.
tRCD	ACT → legal READ/WRITE access to the row being opened	Same bank. AL = 0 for DDR3/DDR4; DDR5 rules and separate BIOS fields are detailed in section 8.2.
tRP	PRE → next legal ACT	Same bank: completion of preparation for a new row.
tRAS	ACT → legal start of PRE	Same bank: minimum active time. This minimum alone does not override post-read/write requirements.
tRC	ACT → next ACT	Same bank: full row cycle, not the activation interval for different banks.

A.6. Changes in interbank constraints and post-write intervals

DDR3 has no bank groups, whereas DDR4 and DDR5 distinguish transitions within a BG from those between BGs. tRRD relates ACT → ACT to different banks of the selected device/rank; tCCD in this example relates READ → READ. For DDR4/DDR5, the constraint for the relevant BG scope is selected; the shared DQ bus and the tFAW window apply simultaneously. Detailed DDR5 conditions appear in chapters 9 and 10.

tWR and tWTR do not begin when WRITE is issued. For full BL8 in DDR3/DDR4 and BL16 in DDR5, in the selected basic mode, the reference is the first rising CK clock edge after the last write data. The final event for tWR is a legal PRE to the same bank; for tWTR, a legal READ within the selected device's scope. In DDR4/DDR5, the BG scope additionally selects the applicable WTR constraint. Write latency and the data itself come first, followed by the corresponding interval.

Fixed BC4 and BC4 with command-selected burst chop in the DDR3/DDR4 considered here differ in the external reference used for write recovery. This is why full bursts are compared here. DDR5 reference events are shown in detail in chapter 11.

tRTP relates the internal READ to permission for PRE in the same bank. In DDR3/DDR4 with AL enabled, the external READ is issued before the internal one, so the interval from external READ to PRE includes AL. The table in section A.5 assumes AL = 0.

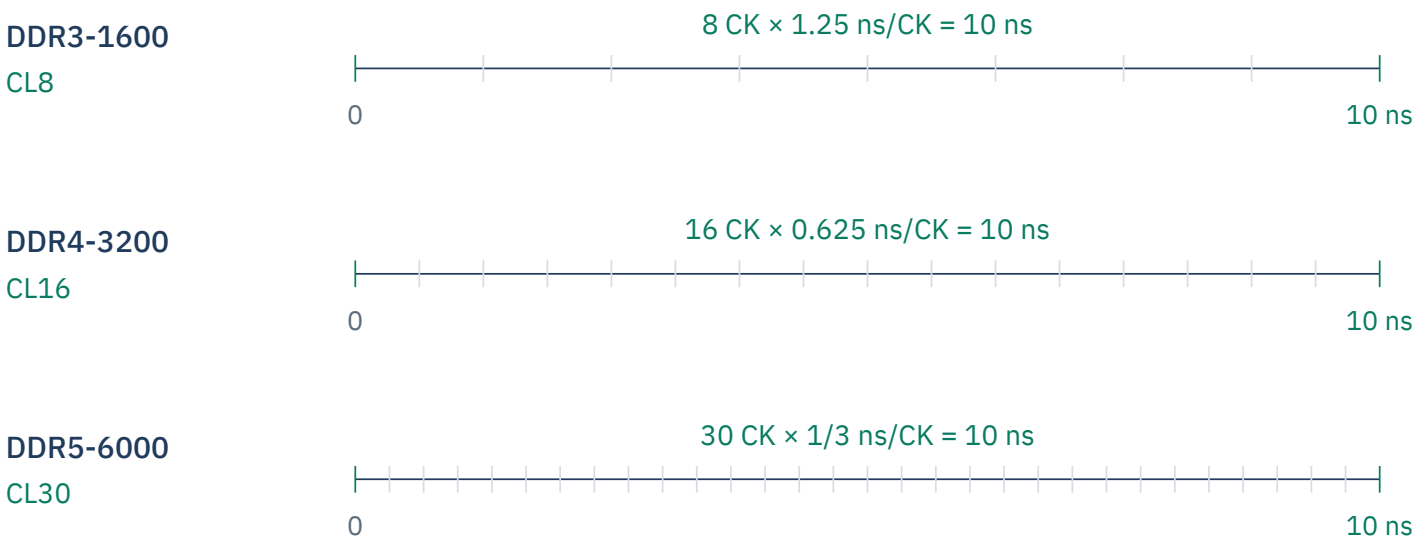
DDR5 has additional constraints for different write sequences and modes; these are discussed in section 10.3. The mapping of these constraints to controller fields is considered in chapter 16.

A.7. CL8, CL16, and CL30: examples of 10 ns CAS latency

The CL number must be compared together with the CK clock period. According to the formula in chapter 4, tCK = 2000 / MT/s in nanoseconds. For the selected examples, DDR3-1600 CL8, DDR4-3200 CL16, and DDR5-6000 CL30, the CAS interval is the same: 10 ns.

These examples were selected for an arithmetic comparison of CAS, not as JEDEC profiles or setting recommendations. The calculation uses the basic mode with AL = 0 for DDR3/DDR4. In the DDR5 considered here, read DQS offset changes the strobe position while preserving the data position and this CL interval.

Figure 32. Different CK counts fit into the same 10 ns



Start: READ reference point. End: nominal start of data.

A shared linear scale of physical time. Illustrative DDR3-1600 CL8, DDR4-3200 CL16, and DDR5-6000 CL30: AL = 0 for DDR3/DDR4, with no additional latency modes; the DDR5 reference point is the second READ cycle. Small divisions denote full CK periods; “ns/CK” means nanoseconds per cycle.

Example mode	One CK cycle	CAS interval	64-byte transfer
DDR3-1600 CL8	1.25 ns	8 × 1.25 = 10 ns	64 DQ, BL8: 4 CK cycles = 5 ns
DDR4-3200 CL16	0.625 ns	16 × 0.625 = 10 ns	64 DQ, BL8: 4 CK cycles = 2.5 ns
DDR5-6000 CL30	1/3 ns	30 × 1/3 = 10 ns	32 DQ, BL16: 8 CK cycles = 8/3 ≈ 2.67 ns

A.8. Four measures of time and transfer rate

The CAS interval ends when data output begins. The block transfer duration describes the burst that follows: with two transfers per CK cycle, it is BL/2 full cycles. Thus, the same 10 ns before data output in the previous example accompanies different transfer times for 64 bytes. Slots are counted in full; the last transfer starts half a cycle before the end of the slot.

Processing of a CPU request begins earlier still: it may include cache misses, queuing, row preparation, and passage through the controller. These stages are not part of CL. Sustained bandwidth, meanwhile, describes a long stream that accounts for program dependencies and pauses for activations, direction changes, and refresh.

In an ideal continuous stream, the useful bandwidth limit in MB/s equals MT/s × number of DQ lines / 8; for GB/s, the result is divided by 1000. In the specified modes, this is 12.8 GB/s for 64-DQ DDR3-1600, 25.6 GB/s for 64-DQ DDR4-3200, and 24 GB/s for one 32-DQ DDR5-6000 subchannel. Two simultaneously fully utilized subchannels provide 48 GB/s. Pauses and request dependencies reduce the bandwidth available to the program; chapter 15 gives more detail.

A.9. Refresh organization

All three generations need refresh. What changes is how maintenance can fit into the stream of work. In the DDR3 considered here, normal REF occupies all banks of the device, while self-refresh retains data in an internal mode. Maintenance frequency depends on temperature and the specific device.

In the DDR4 considered here, Fine Granularity Refresh, which uses shorter, more frequent refresh operations, offers 1×, 2×, and 4× modes: more frequent maintenance with a shorter individual tRFC. The reduction in tRFC need not be proportional to the increase in command frequency, so total maintenance time may not decrease.

The SK hynix DDR5 in this comparison uses normal mode and FGR, with twice the frequency of the corresponding REFab commands at the same temperature. In addition to REFab for all banks, FGR offers REFsb: it selects the same bank number in every BG of the device. The other banks can operate subject to restrictions.

tRFC is measured from the accepted refresh command until permitted subsequent actions within its scope; tREFI describes the permissible average schedule for these commands. DDR4 and DDR5 have different sets of modes and different rules for mixing them. Temperature-related increases in refresh frequency and the choice of FGR also act separately. DDR5 details appear in chapter 12.

A.10. Power, data protection, and interface training

Nominal VDD/VDDQ voltages decrease from 1.5 V for DDR3 to 1.2 V for DDR4 and 1.1 V for DDR5. This reduces one component of switching energy. Module temperature also depends on capacity, device count, data rate, load, power supply, and cooling. VDD supplies the main DRAM circuits, and VDDQ supplies the I/O interface; the DDR4/DDR5 examples use a separate VPP rail for row activation.

A standard DDR3/DDR4 UDIMM receives its required supply voltages from the board. In client DDR5, the main voltages are generated by a PMIC on the module itself: the typical input is 5 V, and VDD/VDDQ/VPP are generated locally.

DDR5 adds built-in on-die ECC. It helps correct errors in the internal array but does not protect the external transfer between the DIMM and the controller. System ECC requires separate support from the memory, controller, and platform. ECC, ECS, and CRC operation is discussed in section 10.5.

Interface training evolved across all three generations. DDR3 supports write leveling, aligning the write strobe to the clock; DDR4 extends interface training, including VrefDQ, the reference voltage for DQ reception. DDR5 includes command/address training, more complex read patterns, and additional signal-reception features. In the SK hynix datasheet, complex patterns are associated with higher transfer rates; DFE, decision feedback equalization, uses feedback from previously received data to help counter intersymbol interference.

Scope	DDR3	DDR4	DDR5
Nominal DRAM supply voltages in the selected documents	VDD/VDDQ: 1.5 V; DDR3L separately: 1.35 V	VDD/VDDQ: 1.2 V; VPP: 2.5 V	VDD/VDDQ: 1.1 V; VPP: 1.8 V
Power for a standard client UDIMM	Rails from the board	Rails from the board	PMIC on the module; typical input 5 V
Internal error correction	The selected datasheet does not specify on-die ECC	In Micron's comparison: no on-die ECC	On-die ECC within DRAM; the external path remains separate
System ECC	Depends on the module, IMC, and platform	Depends on the module, IMC, and platform	Requires separate support despite on-die ECC

> The nominal values in the power and protection table apply to the DRAM rails of the specified memory variants. CPU VDDQ and SA are separate voltages.

A.11. The effect of architecture on memory performance

The memory generation defines the architectural capabilities: bank organization, interface width and count, burst transfer, and maintenance modes. The particular kit determines which of them are available: capacity, device density and width, ranks, MT/s, timings, and voltages all matter here.

The result depends on the entire platform and the request stream. Two DDR5 kits can load the controller differently and use parallelism differently. Comparisons between generations also involve CPU and board differences; an increase in FPS or bandwidth, or a reduction in latency, is established for a specific workload.

The general access logic remains: open a row, meet the constraints, transfer data, and maintain the array on time. DDR5 changes the organization, reference events, and modes that determine the schedule for these actions.

Appendix B. Abbreviations and notation

This appendix collects the notation used in the guide. The links lead to substantive explanations; the conditions for each specific operating mode remain in the relevant section.

B.1. Abbreviations, commands, and registers

Notation	Brief explanation	Definition
1T1C	A DRAM cell model: one access transistor and one capacitor.	section 2.1
3DS	A DRAM organization with stacked dies; constraints may depend on the logical rank within the stack.	section 9.4
ACT / ACTIVATE	The command that activates a row in the selected bank.	section 6.1
AP	Automatic bank closure after a read or write, subject to the conditions that permit PRE.	section 6.3
BC / BC4 / BC8	A shortened burst; the number specifies the external transfers on a DQ line. Fixed mode and command-selected operation differ.	section A.1; section 10.1
BG	A bank group with shared constraints on certain command sequences.	section 3.3
BIOS / UEFI	The motherboard firmware whose menus provide access to settings.	section 1.4
BL / BL8 / BL16 / BL32	The number of transfers on each DQ line in one burst; the byte count also depends on interface width.	section 3.7; section A.1
CA	The command and address interface between the controller and DRAM.	section 6.1
CK	The DRAM interface clock signal; tCK is its period.	section 4.1
CKD	A buffer that distributes the CK clock signal on a CUDIMM or CSODIMM.	section 3.5
Command Rate	A platform command issue and reception mode; its function differs from the frequency ratio set by Gear.	section 4.4
CPU	The processor that executes program instructions and works with program data.	section 1.1
CRC	A check code for detecting transfer errors between the controller and DRAM.	section 10.5
CSODIMM / CUDIMM	SO-DIMM / DIMM modules with a CKD clock buffer.	section 3.5
DCLK	The clock used to express controller intervals in the TC_RFP documentation. It is not automatically equated with CK or QCLK.	section 16.3
dd / dg / dr / sg	Intel suffixes: a different DIMM; a different BG; a different rank on the same DIMM; the same BG.	section 16.1
DDR / DDR3 / DDR3L / DDR4 / DDR5	DDR means two data transfers per CK period; the number identifies the generation. DDR3L is a DDR3 variant with a lower nominal voltage.	section 4.1; section A.10
DFE	A means of suppressing intersymbol interference in the data receiver.	section A.10

Notation	Brief explanation	Definition
DIMM	A memory module containing DRAM devices, interconnections, and supporting components.	section 3.1
DLL	A delay-adjustment circuit; it may require additional time to become ready after a mode exit.	section 13.2
DLR / SLR	A different / the same logical rank in a 3DS organization; the constraint is scoped relative to the initial operation.	section 14.3
DQ / DQS	Data lines / the accompanying strobe whose edges determine when the receiver captures data.	section 4.1
DRAM / SDRAM	Dynamic memory with periodic charge restoration; SDRAM uses a synchronous command interface.	section 1.4
ECC	Error correction using check data; the scope of protection depends on the implementation.	section 3.6
ECS	An internal DRAM sweep that reads data, corrects errors, and stores the corrected result.	section 10.5
FGR	A refresh mode using shorter, more frequent operations; temperature-related increases in refresh frequency are specified separately.	section 12.3
Gear	The ratio between the memory-interface and controller frequencies in the Intel documentation considered here.	section 4.4
I²C / I3C Basic	Two-wire sideband interfaces for communicating with module components.	section 3.5
IMC	The memory controller integrated into the processor; it accepts requests and schedules commands.	section 1.1
IOL / RTL Offset	Data-path adjustment and RTL offset parameters; their full operation is not disclosed by the public RTL definition used here.	section 16.2
MLC	Intel Memory Latency Checker, a program for measuring memory latency at different load levels.	section 15.3
MR / MR0 / MRR / MRW	A DRAM mode register; MR0 is register 0; MRR reads a register, and MRW modifies it.	section 13.3
MRC	Intel memory-initialization code involved in interface configuration and training.	section 7.3
ODECC / on-die ECC	Correction of a single-bit error within a DRAM codeword. In the device described, a normal READ does not write the correction back to the array.	section 10.5
ODT	On-device termination used to match the line and reduce signal reflections.	section 13.4
OREF_RI / TC_RFP	OREF_RI in the TC_RFP register specifies how long a rank must be idle before refresh may be scheduled in that idle period.	section 16.3
OTF	Selection of a supported burst variant by the command itself, on the fly.	section 10.1
PD / PDE / PDX	The reduced-power mode; entry into it; exit from it. PD does not initiate autonomous refresh.	section 13.1

Notation	Brief explanation	Definition
PHY	The physical portion of the memory interface. In the Intel implementation considered here, Gear relates the DDR PHY and QCLK frequencies.	section 4.4
PMIC	An IC that generates the module supply voltages and controls their delivery.	section 3.5
PRAC	Counting activations of each row, with service requested through back-off and RFM commands.	section 12.6
PRE / PREab / PREpb / PREsb	Closing a row and preparing the bank. ab / pb / sb denote all banks / one bank / same-numbered banks in every BG.	section 8.3
QCLK	The controller clock in the Intel documentation considered here; its period is related to tCK through a confirmed Gear ratio.	section 4.4
RAM	Main memory that holds the instructions and data used by a program.	section 1.1
RBL / WBL	The calculated read / write burst length used in datasheet equations. It does not always equal the actual number of BC8 transfers.	section 10.1
RD / READ / WR / WRITE / RDA / WRA	Commands to read / write a portion of the active row; RDA and WRA additionally request automatic bank closure.	section 6.3
REF / REFRESH / REFab / REFsb	Refresh; REFab services all banks in the selected device, while REFsb services the same-numbered banks in every BG.	section 12.4
RFM / RFMsb	Service associated with row activity. RFMsb applies to same-numbered banks in every BG; the selected device requires FGR.	section 12.6
RMW	Internally reading the previous data, merging it with new data, and writing the result during a partial update.	section 10.3
RTL / SC_ROUNDTRIP_LATENCY	The time from READ to the first data reaching the controller; the SC_ROUNDTRIP_LATENCY register expresses it in QCLK cycles.	section 16.2
SPD	Nonvolatile module information; the SPD Hub provides access to it and to supporting components.	section 3.5
SR / SRE / SRX	The autonomous data-refresh mode; entry into it; exit from it.	section 13.2
SSD	A solid-state drive for persistent file storage.	section 1.1
TCASE	The IC case temperature; the module sensor reading may differ from it.	section 12.3
UDIMM	A module without a register that buffers commands and addresses.	section 3.2
VDD / VDDQ / VPP; CPU VDDQ / SA	The first three names refer to DRAM supply rails; CPU VDDQ and SA are separate processor voltages.	section A.10
VrefDQ	The reference voltage used to receive data on DQ.	section A.10
XMP	Profiles containing memory settings; comparisons must consider the entire confirmed operating configuration.	section 15.4
ZQ / ZQCal Start / ZQCal Latch	Calibration of the output drivers and termination; starting calibration and latching its result.	section 13.4

B.2. Timing parameters

Notation	Brief explanation	Definition
AL	Additive latency in DDR3/DDR4; set to zero in the comparative examples.	section A.5
CAS / CL / tCL	The historical name CAS and the delay from the READ reference point to the start of data output in the selected mode.	section 8.1
CWL	The delay from the final WRITE cycle to the first effective rising edge of the write DQS, excluding the preamble.	section 11.2
tACTPDEN / tPRPDEN / tREFPDEN	Minimum intervals from ACT / PRE / REF to permitted PDE entry.	section 13.1
tCA2CS / tMOD / tWRPRE	BIOS field names whose exact correspondence to registers on the platform considered here has not been established.	section 16.3
tCCD_*_WTR	The full WRITE→READ interval, including CWL, the calculated burst, and the interval after the write data; the asterisk denotes the scope.	section 11.3
tCCD_L_WR / tCCD_L_WR2 / tCCD_M_WR	WRITE→WRITE constraints; the variant depends on the address scope, write completeness, and the need for RMW.	section 10.3
tCCD_S / tCCD_M / tCCD_L	READ→READ intervals for different BGs / different banks in one BG / the same bank in the cited SK hynix table.	section 10.2
tCCD_WTRA	An additional WRITE→READ AP bound that allows write recovery to finish before automatic PRE.	section 11.6
tCK	The duration of one CK period; at a nominal data rate in MT/s, it equals 2000 / MT/s nanoseconds.	section 4.1
tCKE / tCKCKEH / tSR / tXPDLL	Historical or platform-specific names; their exact interpretation requires documentation for the corresponding register.	section 13.2
tCKSRX	The CK stability requirement before exiting self-refresh.	section 13.2
tCPDED	The delay before command propagation is disabled on entering power-down.	section 13.1
tFAW	A sliding window covering four counted activations; tRRD and bank-readiness conditions also apply.	section 9.2
tMRD / tMRR / tMRW	The delay after programming a mode register / the interval between register reads / the interval between register writes.	section 13.3
tMRRPDEN / tMRWPDEN	Minimum intervals from a mode-register read / write to PDE.	section 13.1
tOSCO / tOSCOA / tOSCOM	tOSCOA and tOSCOM constrain access to the DQS oscillator result; the mapping of the BIOS field tOSCO is not established.	section 13.4
tPD / tXP	The power-down duration constraint / the interval from exit to permitted commands.	section 13.1
tQCLK	The QCLK period; with the frequency ratio established, tQCLK = Gear × tCK.	section 4.4
tRAS / nRAS	The minimum ACT→PRE interval for one bank / its value in CK cycles; constraints after reads and writes also apply.	section 8.4
tRC / nRC	The minimum ACT→ACT interval for the same bank / its value in CK cycles.	section 8.5

Notation	Brief explanation	Definition
tRCD / tRCDRD / tRCDWR	ACT→READ/WRITE; separate BIOS read and write fields must be matched to device requirements.	section 8.2
tRDPDEN / tWRAPDEN / tWRPDEN	Intervals from a read / write with AP / write to PDE, including data completion and recovery.	section 13.1
tRDRD_* / tRDWR_* / tWRRD_* / tWRWR_*	Intel fields for read→read / read→write / write→read / write→write transitions.	section 16.1
tREF	The full array refresh window. In the basic Normal REFab example at 0–85 °C, it is 32 ms.	section 12.2
tREFI / tREFI1 / tREFI2	The maximum average refresh interval; indices 1 and 2 refer to normal mode and FGR.	section 12.3
tREFIx9	The TC_RFP field that bounds an interval under Intel controller rules; its unit is 1024 DCLK.	section 16.3
tREFSBRD	The interval from REFsb to permitted ACT in the other banks.	section 12.4
tRFC / tRFC1 / tRFC2 / tRFCsb	The minimum pause after refresh; the variant depends on the command, mode, and die density.	section 12.3
tRFC*_dlr	A transition to refresh of a different 3DS logical rank; it does not complete service in the original rank.	section 14.3
tRFCpb	A BIOS or utility field name; the name alone does not establish its correspondence to DDR5 tRFCsb.	section 12.4
tRFM	The duration of additional RFM service.	section 12.6
tRP / nRP / tRPab	PRE→ACT in the same bank / its value in CK cycles; the separate name tRPab requires a verified mapping.	section 8.3
tRRD_S / tRRD_L	ACT→ACT in different banks: in different BGs / within the same BG.	section 9.1
tRTP	READ→PRE for the same bank; CL and burst duration are not added to it separately.	section 11.5
tRTW / tRDWR	Names for the READ→WRITE transition; the calculation depends on the level of representation, mode, and interface conditions.	section 11.4
tWR	From the first rising CK edge after the last write data to permitted PRE in the same bank.	section 11.5
tWTR / tWTR_S / tWTR_L	The interval from the write-end reference point to permitted READ; the scope is selected by bank location.	section 11.3
tXS / tXS_DLL / tDLLK	Waits after SRX for commands that do not require / do require a ready DLL; tDLLK specifies the DLL settling time.	section 13.2
tZQCAL / tZQLAT	The duration of background ZQ calibration / the interval after its result is latched.	section 13.4
tZQCSPeriod / tZQOPER	BIOS fields whose exact register mappings are not established in this guide.	section 13.4

B.3. Units of measurement and notation

Notation	Brief explanation	Definition
1R / 2R / 1Rx8 / 2Rx8	One / two ranks; x8 additionally specifies the device interface width.	section 3.2
8n / 16n	Internal prefetch of eight / sixteen bits per external DQ line.	section 3.7
b / bit / B	Bit / bit / byte; eight bits form one byte.	section 1.2
ceil / max	Rounding up to an integer / selecting the largest value.	section 4.3
CKn	The point n CK cycles after the chosen time origin.	section 6.2
Gb / Gbit	Density notation in bits; in the DRAM organization considered here, 32 Gbit corresponds to 32×2^{30} bits.	section 1.2
KiB / MiB / GiB	KiB / MiB / GiB: 2^{10} / 2^{20} / 2^{30} bytes.	section 1.2
MHz	Millions of signal periods per second.	section 4.1
MT/s	Millions of transfers per second on one data line.	section 4.1
nCK	The number of CK clock periods.	section 4.1
x4 / x8 / x16	Four / eight / sixteen DQ data lines on one DRAM device.	section 3.2
GB/s	Billions of bytes per second; a decimal unit of bandwidth.	section 1.2
ns / µs	A nanosecond is one billionth of a second; a microsecond is one millionth.	section 1.2

Sources and Documentation

Device requirements come from datasheets, register definitions from Intel documentation, and general principles from research papers. The full restricted-access JEDEC standard was not studied directly; no new hardware measurements were made.

DRAM architecture and units

• NIST. Prefixes for binary multiples. KiB, MiB, GiB, and decimal units. Checked October 7, 2026.

• Adaptive-Latency DRAM. HPCA 2015. §§ 2.1–2.2: the cell, sensing, charge restoration, and precharge.

• A Case for Exploiting Subarray-Level Parallelism in DRAM. ISCA 2012. § 2: banks, subarrays, and the row buffer.

• David T. Wang. Modern DRAM Memory Systems. 2005. §§ 4.4.1–4.4.2: activation load and the FAW window.

DDR3, DDR4, and DDR5

• Micron. 2Gb: x4, x8, x16 DDR3 SDRAM. Rev. S, 02.2016. Organization, commands, burst transfers, power supply, and refresh. hstatic copy.

• Micron. 4Gb: x4, x8, x16 DDR3L SDRAM. Rev. R, 09.2018. p. 1: DDR3L power supply. LCSC copy.

• Micron. 8Gb: x4, x8, x16 DDR4 SDRAM. Rev. M, 10.2017. Organization, bank groups, commands, burst transfers, and refresh. DigiKey copy.

• Micron. TN-40-40: DDR4 Point-to-Point Design Guide. Rev. B, 10.2016. 8n, bank groups, and the electrical interface. oldfriend.url.tw copy.

• SK hynix. 32Gb DDR5 SDRAM, Rev.1.2, 07.2025. Organization, commands, timings, refresh, ECC, and training. UTTC copy.

• Micron. DDR5 DRAM. Architecture and a DDR4/DDR5 comparison.

Modules and supporting components

• Kingston. DDR5 Memory Standard. MKF-954, 2022. pp. 2–3: subchannels, 16n, power supply, and on-die error correction.

• Micron. MTC16C2085S1UC UDIMM Addendum. Rev. F, 10.2021. pp. 1, 7: organization of an ordinary 2Rx8 UDIMM. DigiKey copy.

• Kingston. KVR16N11/8. Datasheet VALUERAM1445-001.D00: a 64-bit 2Rx8 module.

• Kingston. KVR16N11/8 product listing. Classification as a DDR3 UDIMM without system ECC.

• Kingston. KVR26N19S8/8. Datasheet VALUERAM1595-001.A00: a 64-bit 1Rx8 module.

• Kingston. KVR26N19S8/8 product listing. Classification as a DDR4 UDIMM without system ECC.

• Rambus. DDR5 SPD Hub with Internal Temperature Sensor. SPD, the management bus, and the temperature sensor.

• Rambus. DDR5 Client PMICs. Voltage generation and module power management.

• Rambus. DDR5 Client Clock Driver. The role of the CKD in CUDIMM/CSODIMM.

• NXP. I²C-bus specification, UM10204, Rev.7.0. Two-wire bus: SDA and SCL.

• MIPI Alliance. MIPI I3C Basic. Management communication and in-band notifications.

• Kingston. DDR5 Memory Standard: On-die ECC. The scopes of internal and system error correction.

• Kingston Glossary. CAS Latency / CL. Expansion of CAS.

Intel memory controller

• Intel. Architectural Background: address translation. ID 775099, Rev. 2.0: program and physical addresses; without relying on the IMC's undisclosed address mapping.

• 13th Gen Intel Core. SC_ROUNDTRIP_LATENCY, E020. Vol. 2, ID 767624, Rev. 1.3, July 13, 2023: RTL in QCLK.

• 13th Gen Intel Core. MC_BIOS_REQ, 5E00. Vol. 2, ID 767624, Rev. 1.3, July 13, 2023: the GEAR field and the DDR PHY/QCLK ratio.

• 13th Gen Intel Core. TC_RDRD, E00C. Vol. 2, ID 764981, Rev. 1.3: read → read.

• 13th Gen Intel Core. TC_WRRD, E014. Vol. 2, ID 764981, Rev. 1.3: write → read.

• 13th Gen Intel Core. TC_RDWR, E010. Vol. 2, ID 764981, Rev. 1.3: read → write.

• 13th Gen Intel Core. TC_WRWR, E018. Vol. 2, ID 767624, Rev. 1.3: write → write.

• 13th Gen Intel Core. TC_RFP, E438. Vol. 2, ID 764981, Rev. 1.3: tREFIx9 in 1024 DCLK; OREF_RI in DCLK.

Scheduling, latency, and reliability

• Rixner et al. Memory Access Scheduling. ISCA 2000. §§ 1–3: command scheduling and locality.

• Intel. Memory Latency Checker. Measuring idle and loaded latency, as well as bandwidth.

• Vittal, Qureshi. BARD. 2025. § I and § II-D: grouping writes and the wait for reads.

• Patel, Oliveira, Mutlu. HARP. 2021. § 1 and § 4.2: observing errors after on-die correction.

• Canpolat et al. Understanding the Security Benefits and Overheads of Emerging Industry Solutions to DRAM Read Disturbance. 2024. Abstract, v3: PRAC, pausing commands, and RFM.