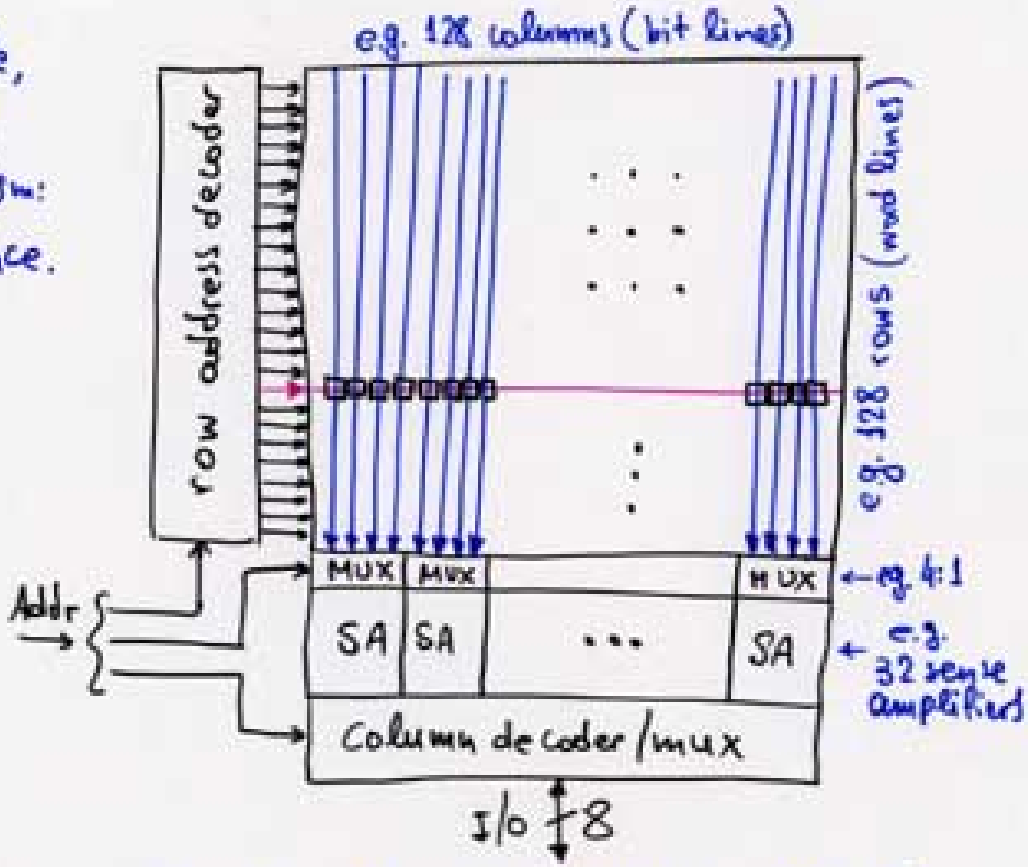
2.1 Buffer Memory Technology

- Memory Blocks On-Chip
  - On-chip SRAM area, access rate, power consumption
- Power Consumption for chip-to-chip communication
- Memory Chips (commercially available)
  - Chip periphery interface: communication standards to memory chips and their off-chip throughput
  - DRAM chips, internal banks, Bank Interleaving

On-Chip SRAM

Memory blocks inherently provide, on-chip, very high throughputs, owing to their inherent parallelism: an entire row is accessed at once. This high throughput is available on-chip, due to the feasibility of very wide datapaths, running at high clock rates.

(Very wide & very large memories are made of several smaller memory blocks, to reduce capacitive loading on word lines and bit lines)

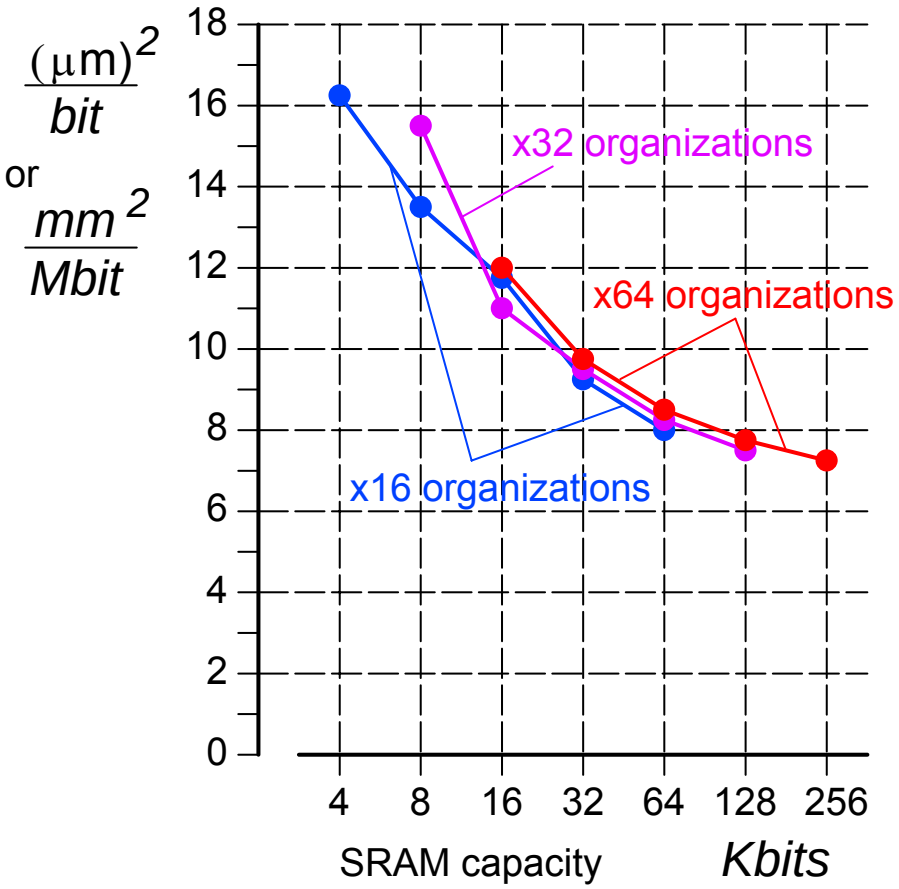


Example layout: 16 Kbit = 2K x 8

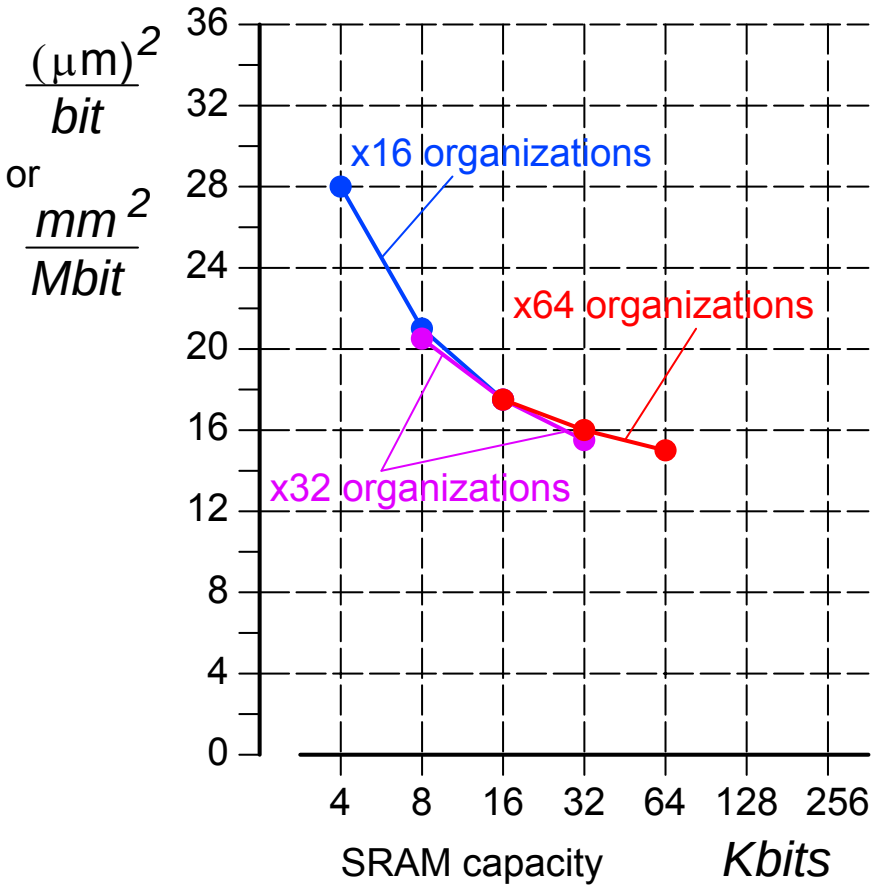
On-Chip SRAM Area per Kbit

(0.18-micron CMOS technology example - 2002)

One-Port SRAM

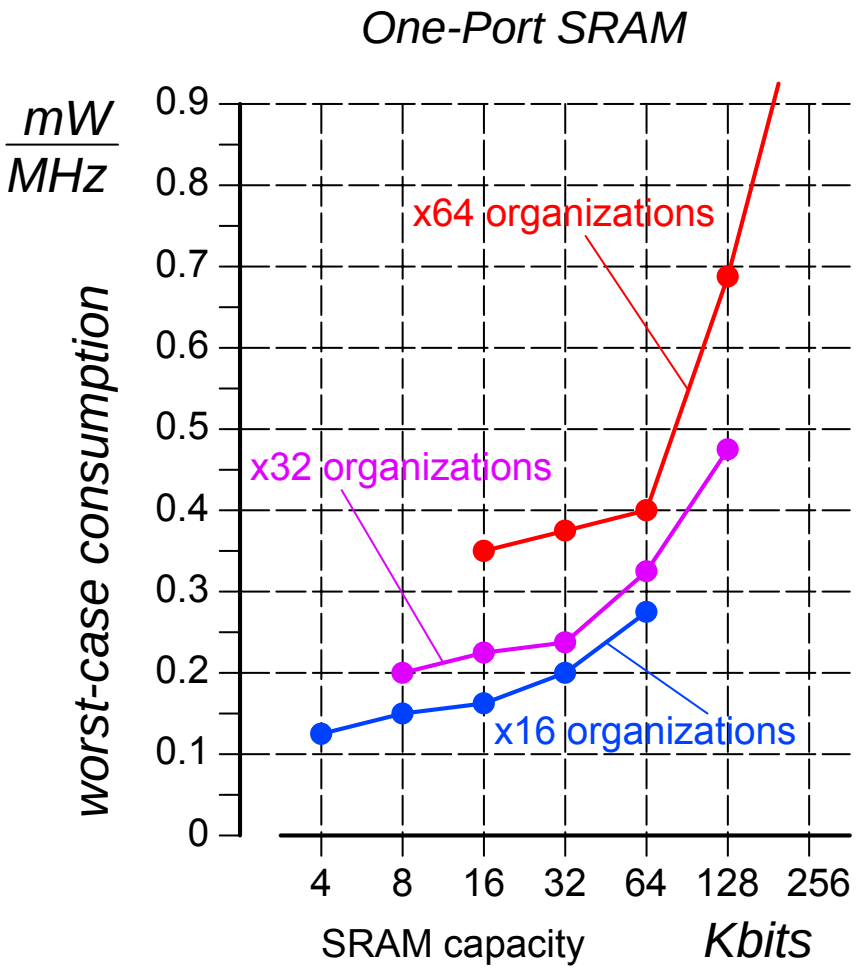


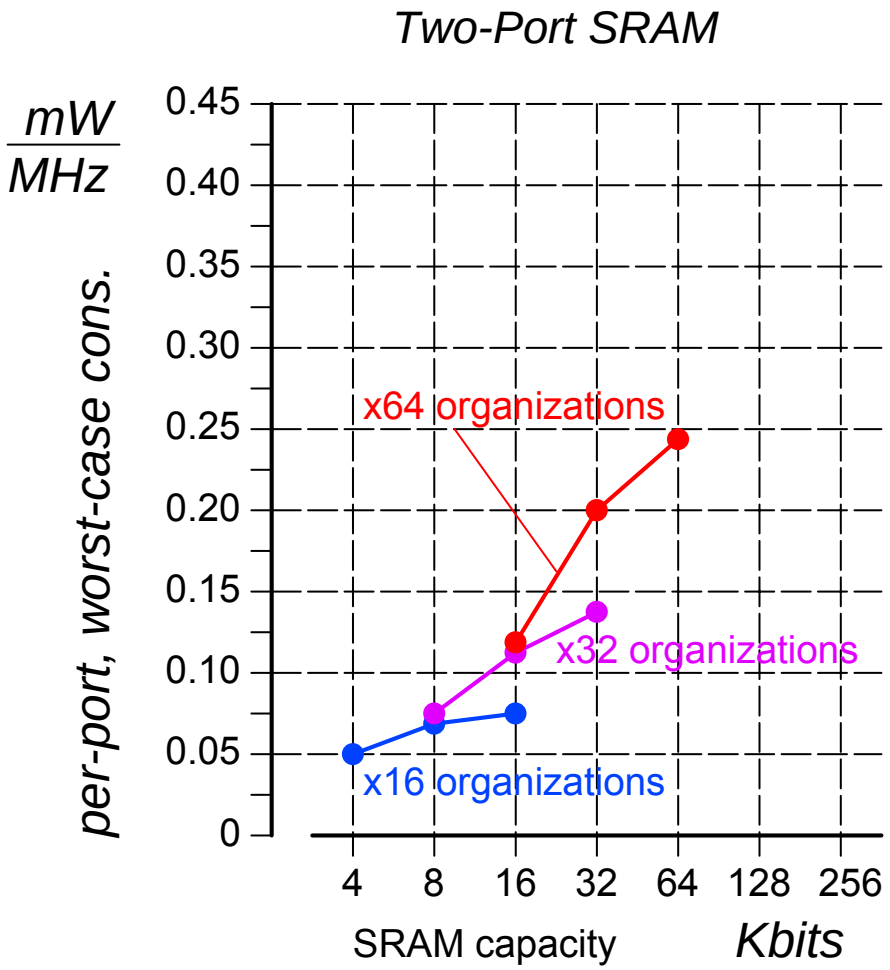
Two-Port SRAM



On-Chip SRAM Power Consumption

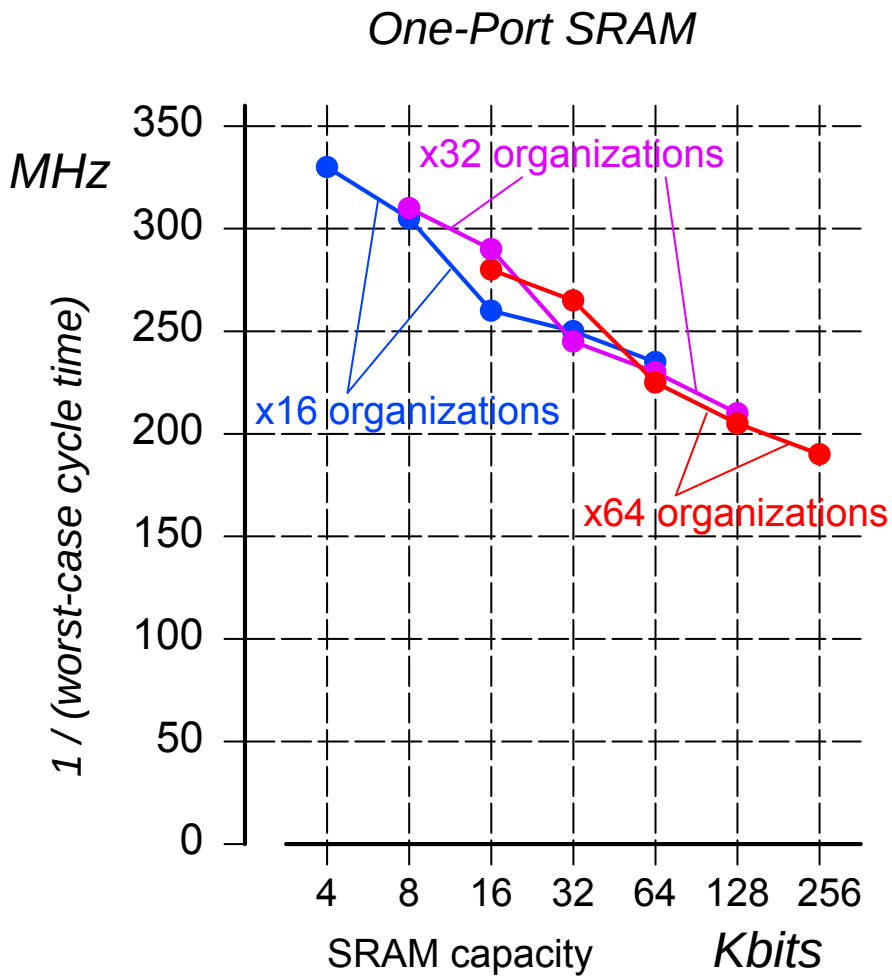
(0.18-micron, 1.8 Volt CMOS technology example - 2002)

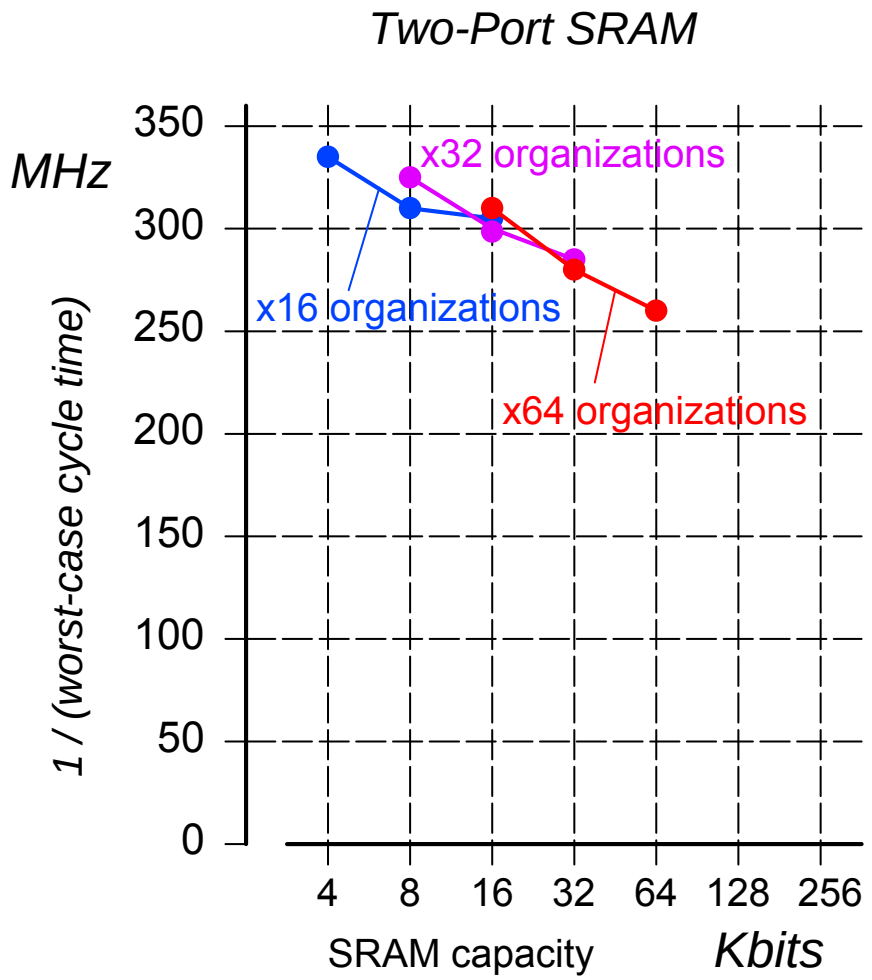




On-Chip SRAM Performance: Access Rate

(0.18-micron, 1.8 Volts CMOS technology example - 2002)





On-Chip SRAM Buffer Example *(i):* 40-Byte wide

- Width = 1 min-size IP packet =

  = 40 Bytes = 320 bits = 5 blocks × 64 bits/block
- One-port, 2048 packets × 40 B = 80 KB = 640 Kb
- 0.18-μm CMOS
- Area: 5 banks × 128 Kb/bank × 7.5 mm^2/Mb =

  = 0.64 Mb × 7.5 mm^2/Mb = **4.8 mm^2**
- Throughput: 320 bits × 200 Macc/s = **64 Gb/s**
- Power Consumption:

  5 banks × 0.7 mW/MHz × 200 MHz = **0.7 W**

On-Chip SRAM Buffer Example *(ii):* 256-Byte wide

- Width ≈ 1 average-size IP packet =

  = 256 Bytes = 2048 bits = 32 blocks × 64 b/bl
- Two-port, 1024 packets × 256 B = 256 KB = 2 Mb
- 0.18-μm CMOS
- Area: 32 × 64 Kb × 15 mm^2/Mb = 2 M × 15 = **30 mm^2**
- Throughput: 2 ports × 2048 b/port × 260 MHz ≈

  ≈ **1 Tb/s** (!) (500 Gb/s writes + 500 Gb/s reads)
- Power Consumption:

  32 banks × 2 ports × 0.25 mW/MHz × 260 MHz = **4.2 W**
- Conclusion: “no problem” on-chip, except for small packets

Power Consumption to Throughput Ratio (1 of 2)

- (1) On-Chip Buffer Memories:
- 0.18-μm CMOS:
  - 1-port, ×16: ≈ 0.15 mW/MHz = 0.15 mW / 16 Mbps = 9.4 mW/Gbps
  - 1-port, ×32: ≈ 0.25 mW/MHz = 0.25 mW / 32 Mbps = 7.8 mW/Gbps
  - 1-port, ×64: ≈ 0.40 mW/MHz = 0.40 mW / 64 Mbps = 6.2 mW/Gbps
- Conclusion: 5 to 10 mW / Gbps on-chip buffer memories
- (2-port memories seem to offer lower consumption/Gbps)

Power Consumption to Throughput Ratio (2 of 2)

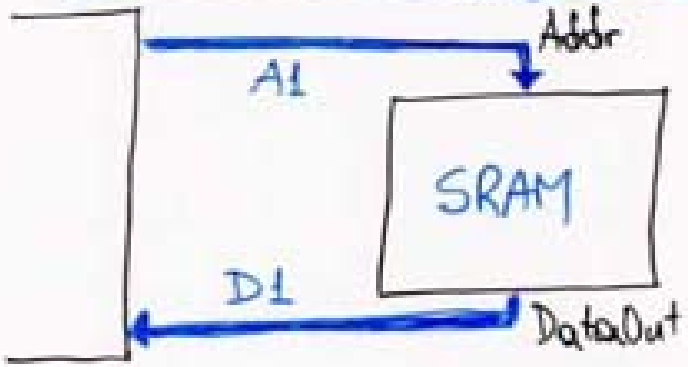
- (1) Chip-I/O Pin Power Consumption:
- both directions of a high-speed serial off-chip transceiver (without equalization, which consumes considerably)
- 0.18-μm CMOS: 25 to 40 mW / Gbps chip-to-chip comm
- 0.11-μm CMOS: ≈ 15 to 25 mW / Gbps
- copper cable power consumption is very small, compared

⇒ Chip-to-chip communication costs much more than buffering

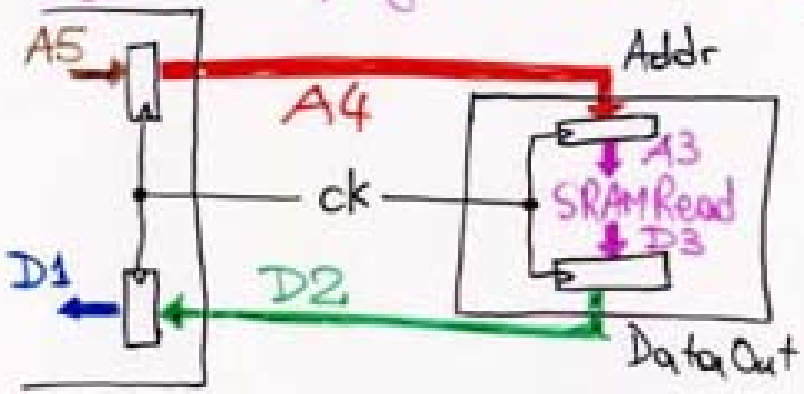
- Total chip power limit (few tens of Watts) limits throughput!

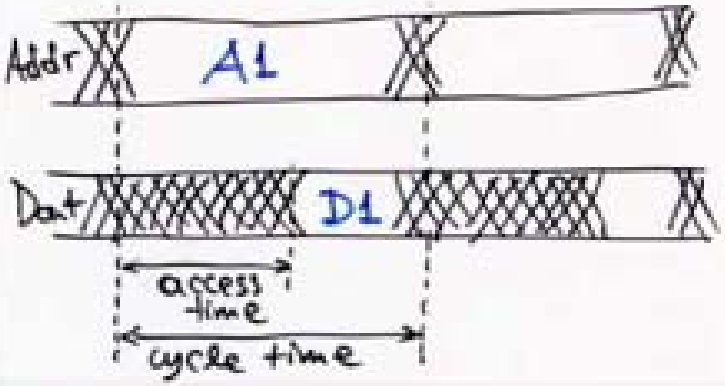
Off-Chip Memory -or other networking/I/O chips: How to Increase Chip-to-Chip Communication Throughput?

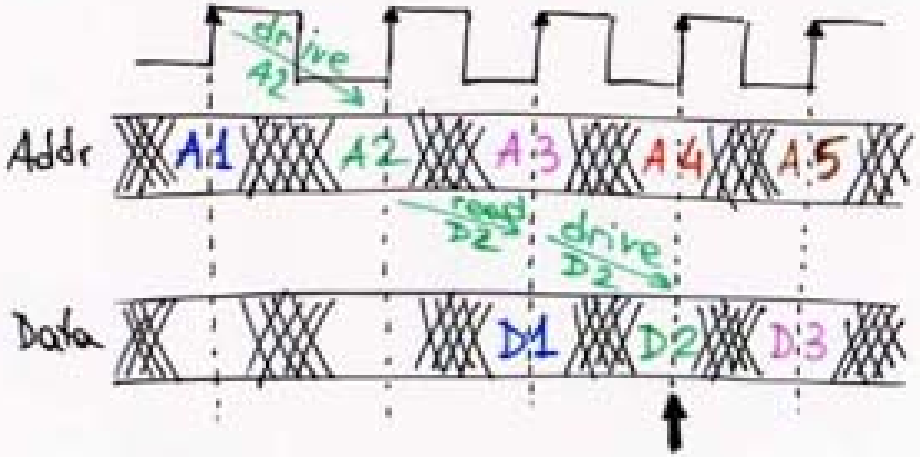
Old SRAM Read ("flow through"):



(1) Pipelined Reads (Synchronous, Registered Interface)



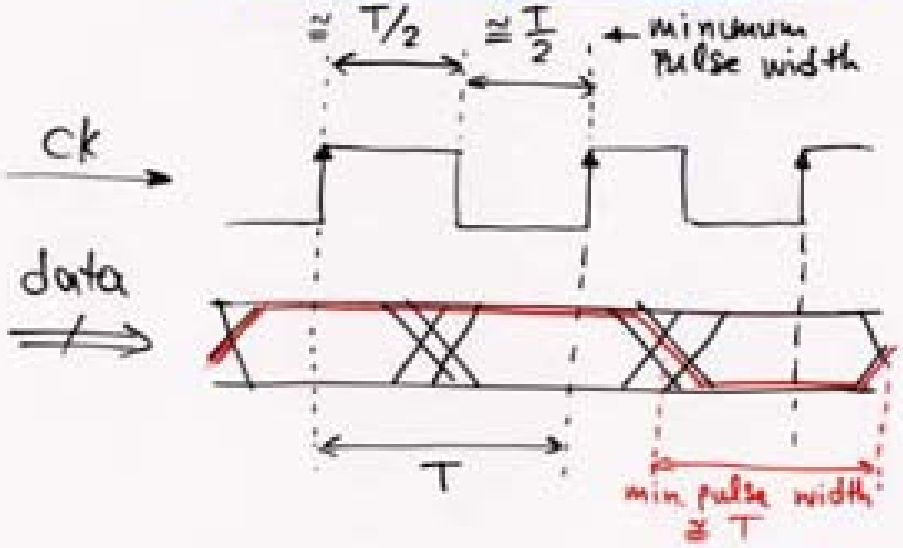




...Further Increasing the data pin through put of chip-to-chip communication:

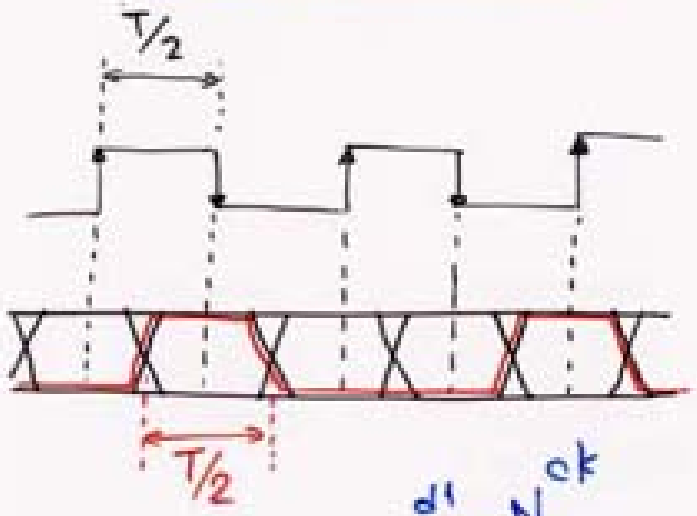
(2) DDR (Double Data Rate) Timing

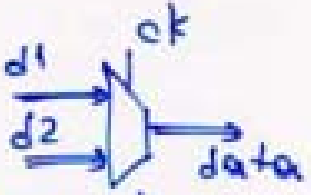
Traditional Synchronous Intf:



Transmit and receive with a positive-edge-triggered register

DDR Interface:



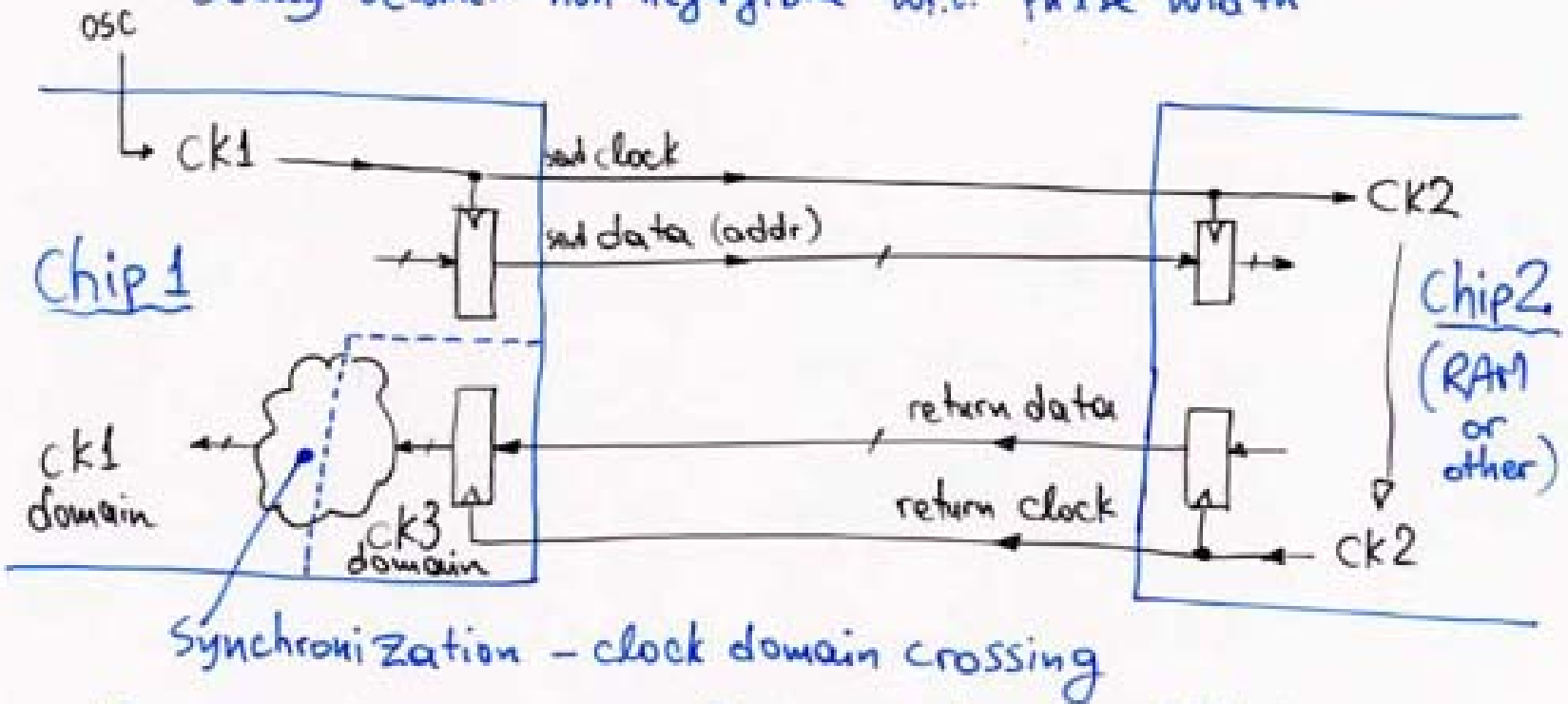


Transmit with:
Receive with: two registers:
- one positive-edge-tr. register
- one negative-edge-tr. register

... further increasing the data pin throughput of chip-to-chip communication...

(3) Source-Synchronous Data Clocking

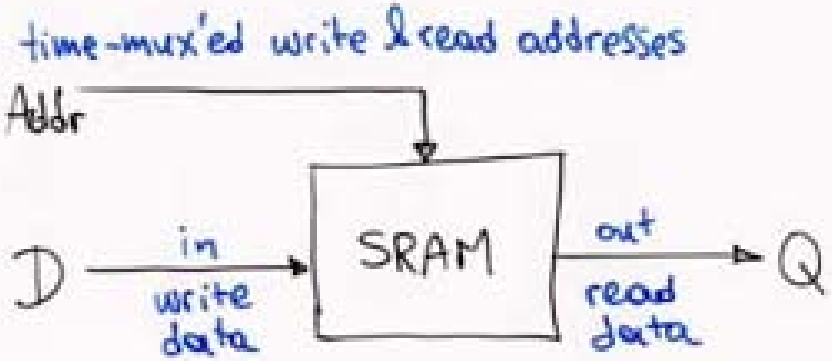
when the clock frequency rises, the chip-to-chip (speed-of-light) delay becomes non-negligible w.r.t. pulse width



ck3 is a delayed version of ck1, i.e. has (exactly) the same frequency, but its delay (phase shift) may vary (slowly) with time...

SRAM Data I/O Paths:

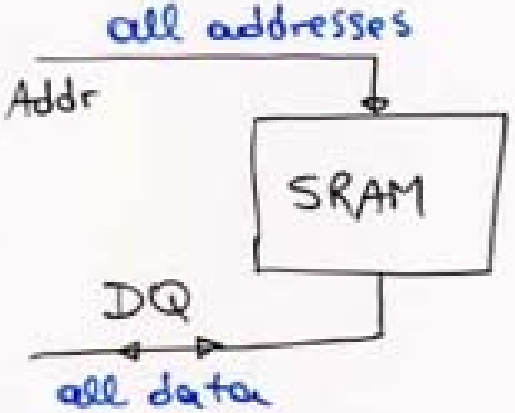
Separate D (in) and Q (out) Paths:



⊖: data path underutilization when inbalanced (≠ 50% - 50%) read/write transactions

Versus

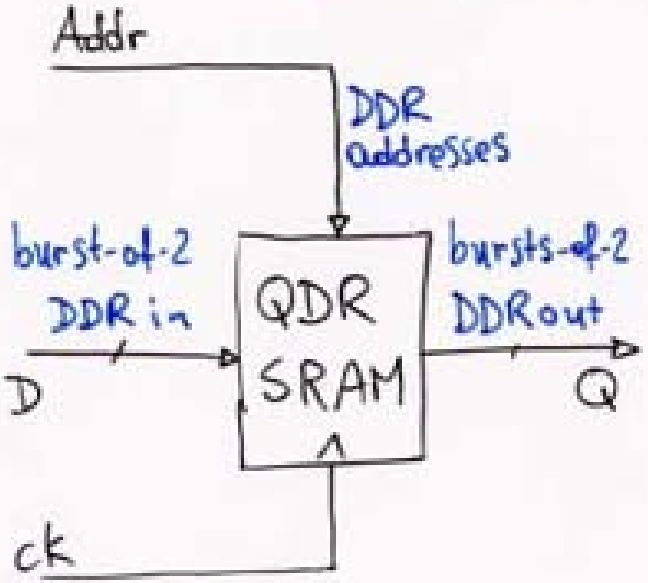
Shared "DQ" Data Bus:

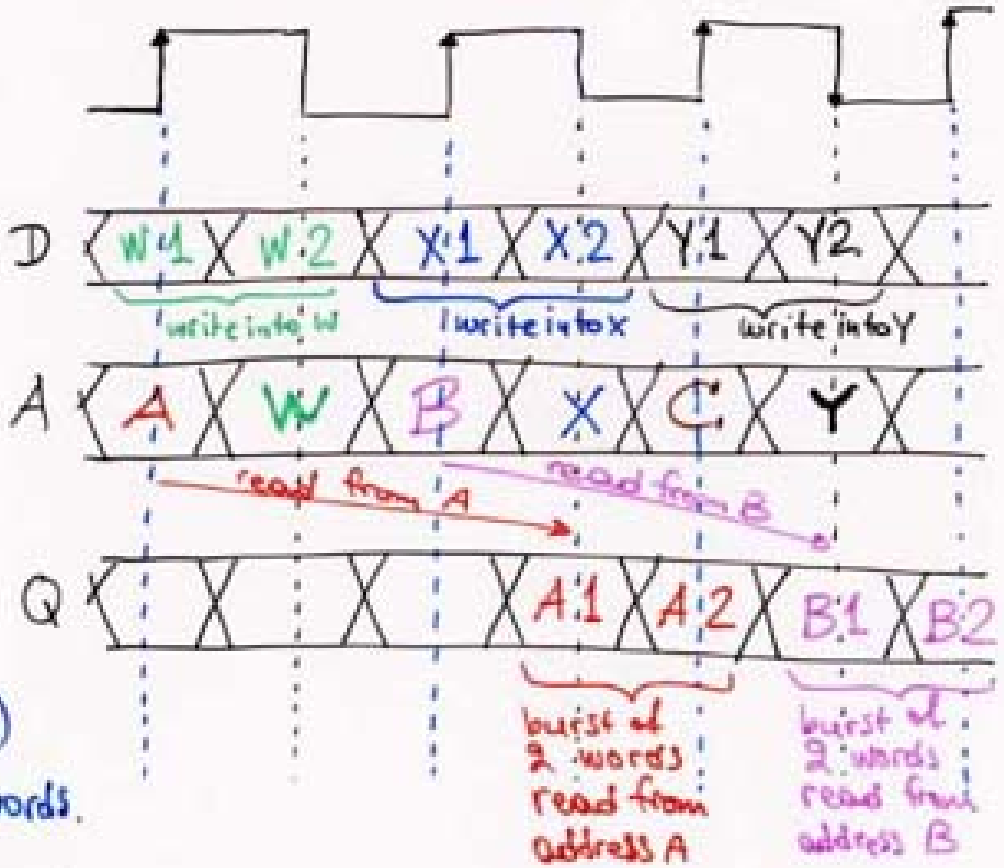


⊖: bus turn-around overhead: data bus underutilization when frequently switching between read & write transactions

modern SRAM chip technology w. separate D(in) & Q(out) paths:

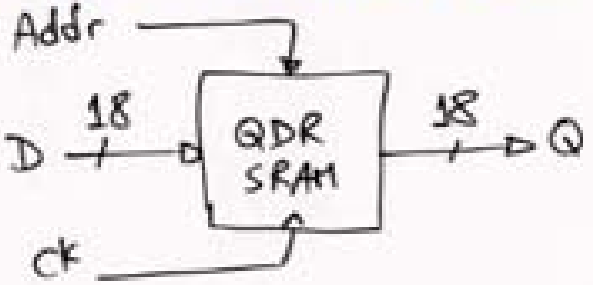
"QDR" (Quad Data Rate) SRAM





Other Version:
"burst-of-4":
- addr. path is plain (NOT DDR)
- each addr. refers to 4 data words.

Example QDR SRAM (2001) Micron's MT54V512H18



9 Mbits = 512 k x 18 bits

Clock freq. up to 167 MHz

$T \geq 6ns$ pulse, bit width $\geq 3ns$

peak write throughput = 167 MHz x 2 (DDR) x 18 bits = 6 Gb/s /chip

peak read throughput = 167 MHz x 2 x 18 bits = 6 Gb/s /chip

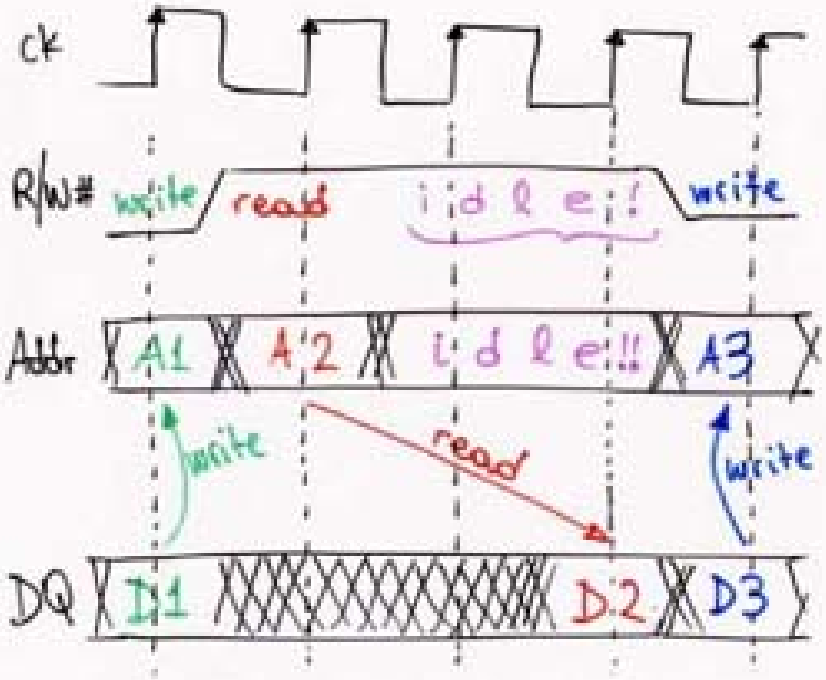
Peak total throughput, when fully balanced 50-50 reads/writes } = 6 + 6 = 12 Gb/s /chip

2.5 Volt power supply; Power Consumption $\cong$ 1 Watt @ 167 MHz

$\Rightarrow$ power per throughput = $\frac{1\ W}{12\ Gbps} \cong 0.08\ \frac{Watt}{Gbps}$

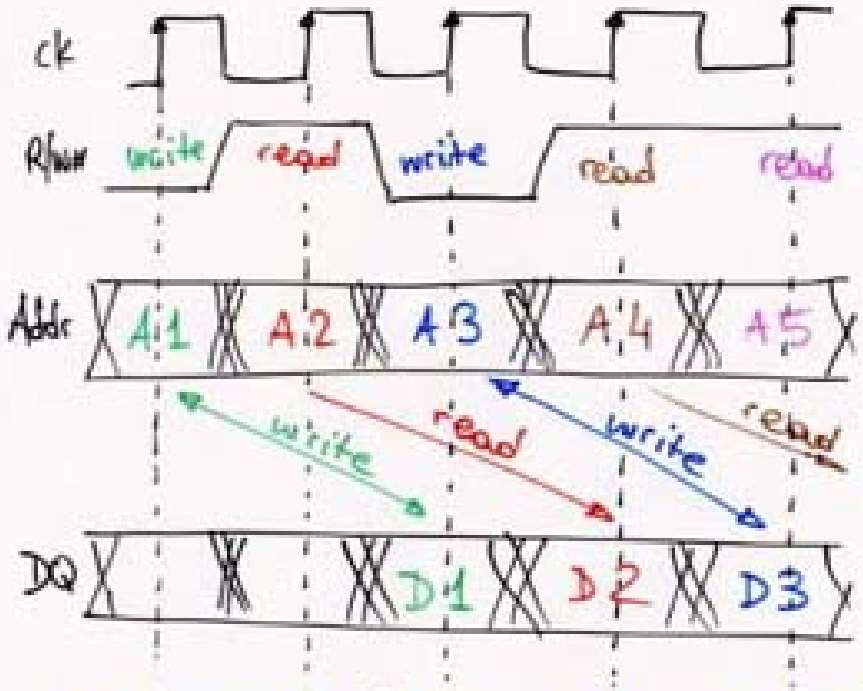
Shared "DQ" Data Bus Timing:

Naive Timing:



Underutilization on every read-to-write transition

"ZBT" (Zero Bus Turn-around) Timing:



D1 has not yet been written at M[A1] when reading from M[A2] starts...
...need to bypass mem. when A2==A1

Example Shared Bus SRAM at the top current performance (2001):

Micron's MT57V256H36 DDR SRAM

9 Mbits = 256K x 36 bits

Clock freq. up to 300 MHz (!) ⇒

T ≥ 3.3 ns, bit pulse width ≥ 1.6 ns

Burst-of-4 accesses only

(one address every 2 clock cycles)

Peak Throughput = 300 MHz × 2 (DDR) × 36 b = 21.6 Gb/s

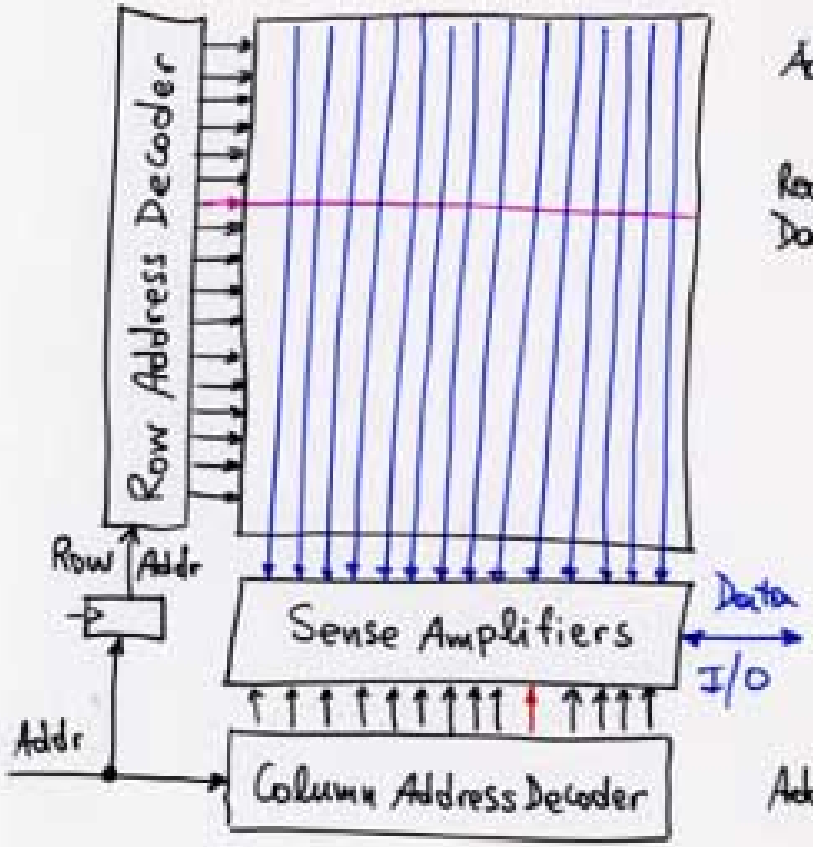
Throughput with alternating read/writes = $\frac{2}{3}$ × peak = 14.4 $\frac{\text{Gbps}}{\text{chip}}$

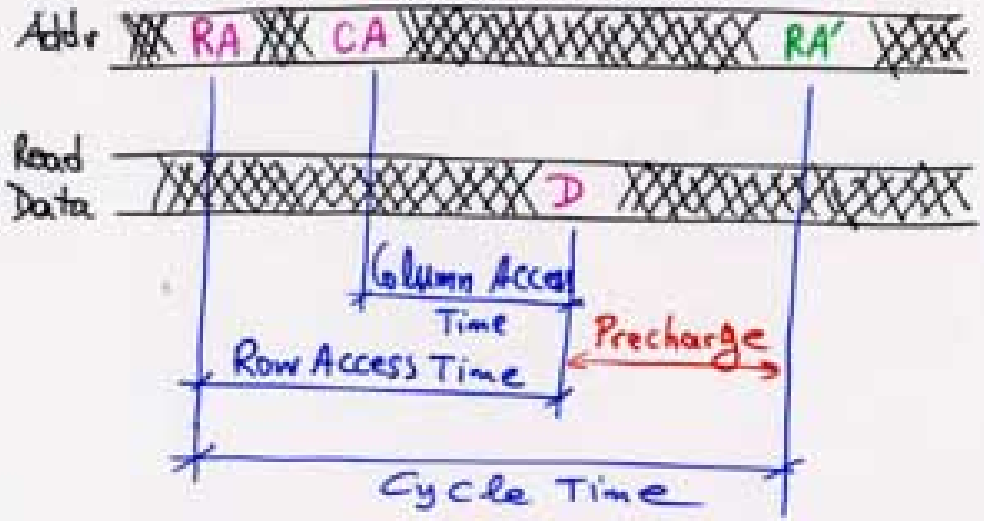
2.5 Volts Power Supply; Consumption = 1.6 W

⇒ ~ 0.1 $\frac{\text{Watt}}{\text{Gbps}}$

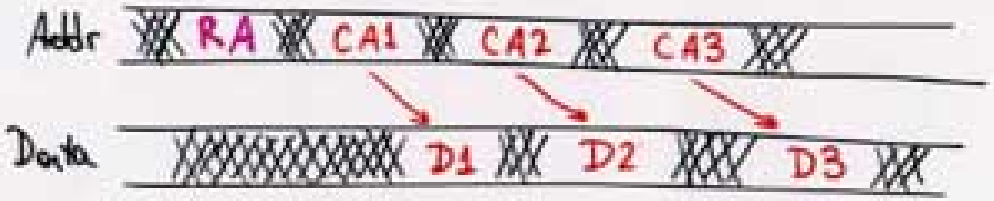
Although the ZBT concept is used, due to the high clock frequency and the unavoidable bus turnaround overhead (multiple drivers on the same wire, each using its own clock (source-synchronous timing)), 1 to 2 clock cycles (= 2 to 4 word burst) are lost on every read-to-write transition.

DRAM Basics: Row Address, Column Address, Precharge





Multiple accesses within same row are faster:



Fast DRAM Example (2001)

Micron MT46 V2 M32

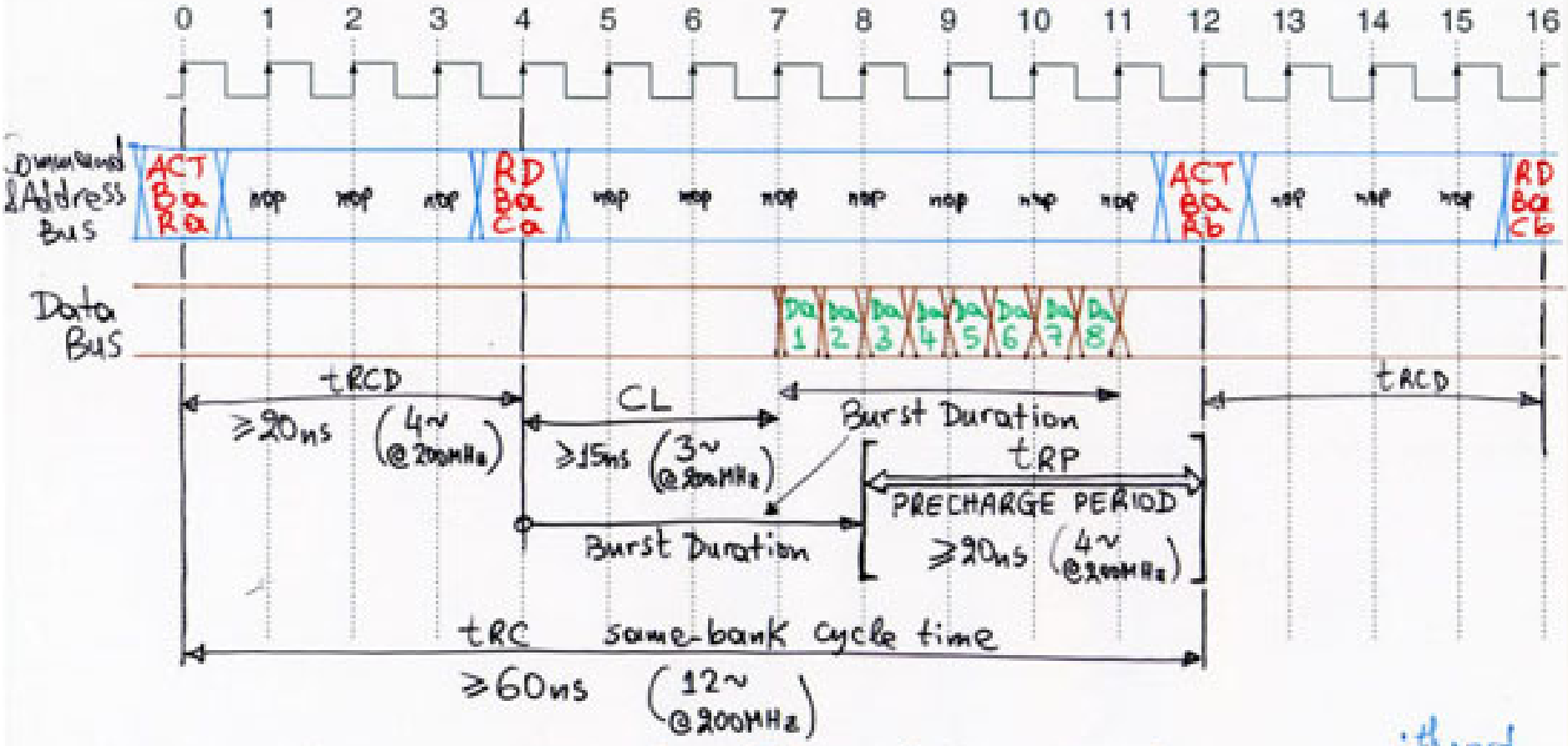
DDR SDRAM

(Synchronous DRAM)

- 200 MHz max. clock frequency
- 64 Mbits = 2M x 32 bits =
  = 512k x 32b x 4 Banks
- 32-bit (shared DQ) databus, DDR timing ⇒
  ⇒ 2 words x 32 bits each per clock cycle
  peak databus throughput
- ≈ 1 Watt at peak access rate, using one bank only, 2.5 Volt. (No number given for multibank op.)

- Row Address-to-Column Address: ..... $t_{RCD} \geq 20ns$ (@200MHz: 4~)
- Column Address-to-Read Data (CAS latency): ... $CL \geq 15ns$ (@200MHz: 3~)
- Write Recovery Time (write data-to-precharge): ... $t_{WR} \geq$ ..... 2~
- Precharge Time: ..... $t_{RP} \geq 20ns$ (@200MHz: 4~)
- Cycle Time (same bank): ..... $t_{RC} \geq 60ns$ (@200MHz: 12~)
- Bank-to-Bank Activation (other bank Row-to-Row): t_{RRD} ..... 2~
- Read-to-Write bus turn-around lost cycles: ..... 3~
- Write-to-Read same bank lost cycles (write recovery time): ..... 2~
- Write-to-Read other bank lost cycles: ..... 0~

Single-Bank Read Access

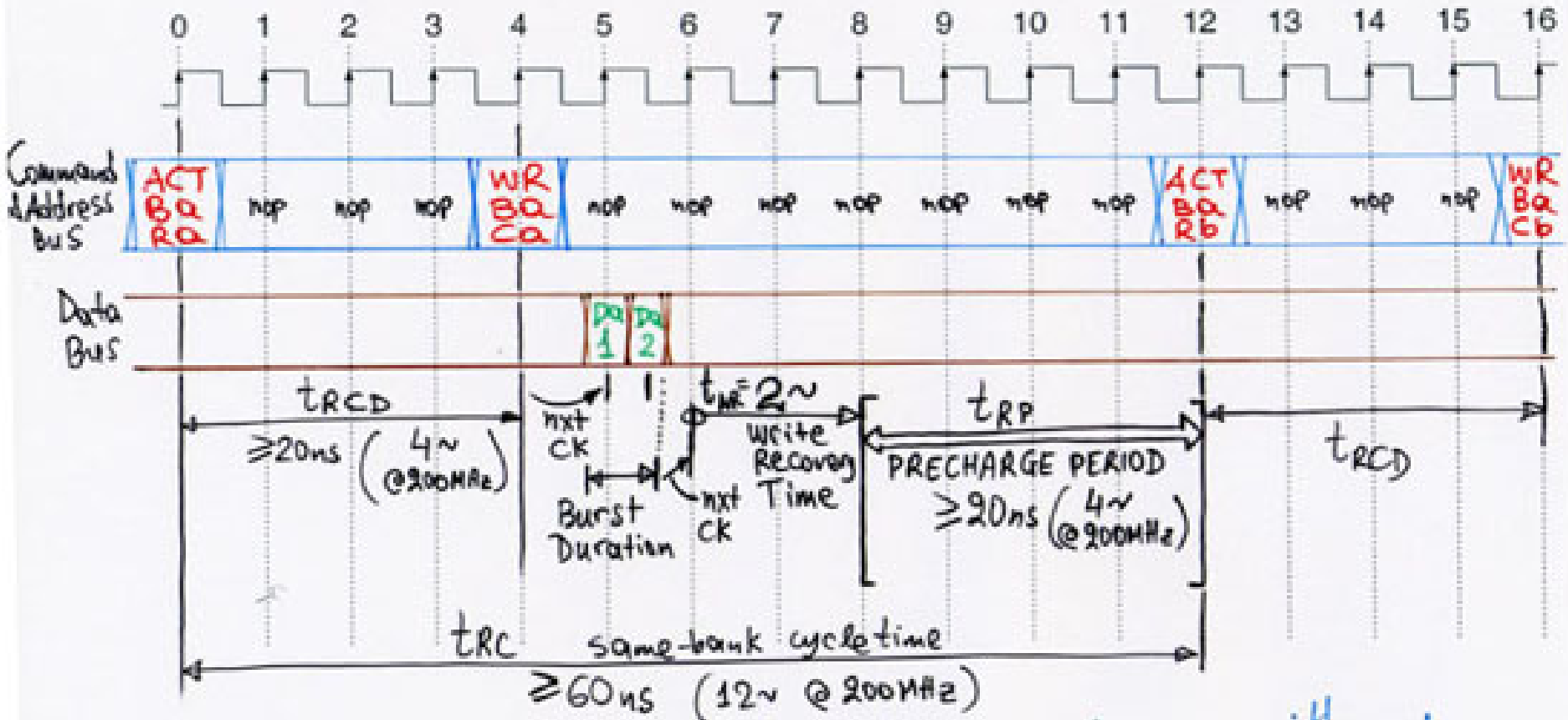


ACT = Activate
Ba = Bank #a
Ra = Row # Ra Address

RD = Read (the predefined burst size)
Ba = from the active Row within Bank #a
Ca = at Column Address #Ca

Di = i-th word of burst from Ba, Ra, Ca

Single-Bank Write Access

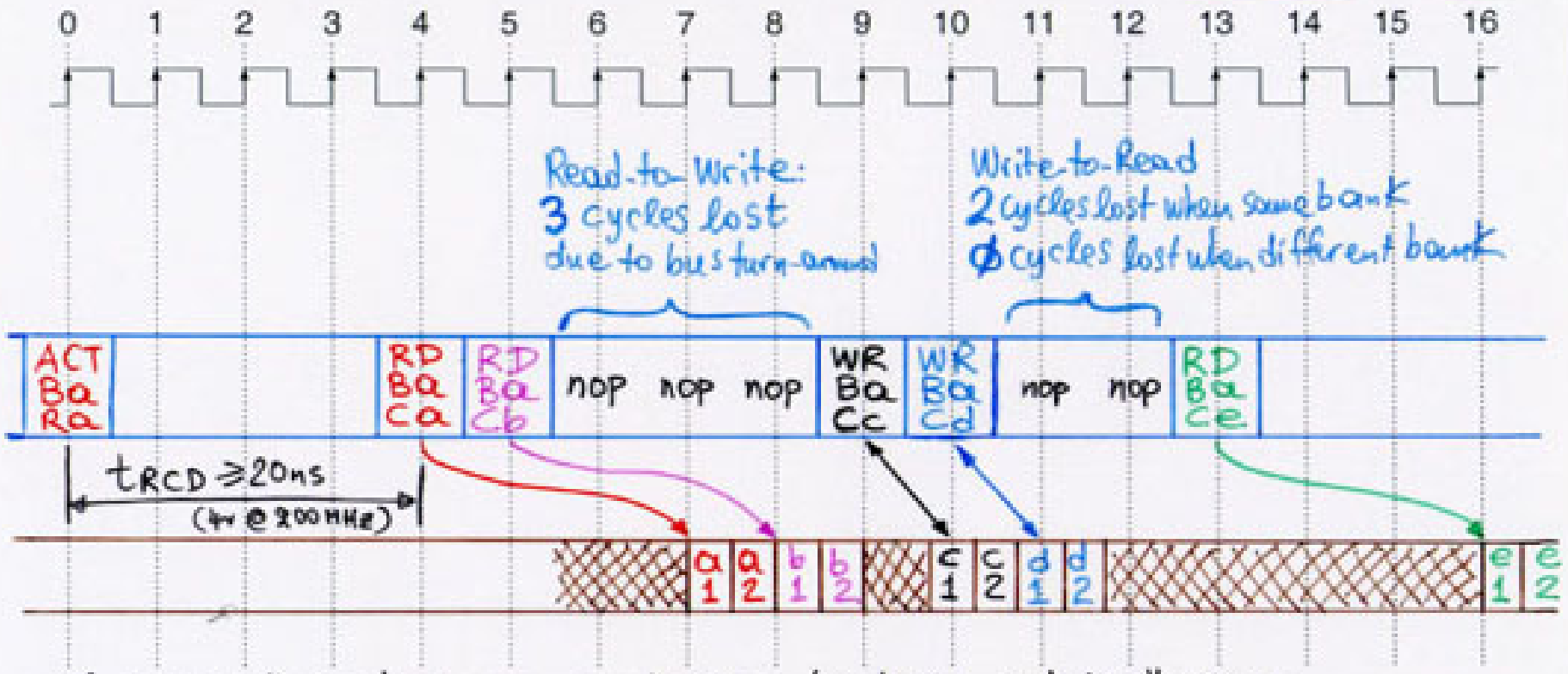


ACT = Activate
Ba = Bank #a
Ra = Row Address Ra

WR = Write (the predefined burst size)
Ba = into the active Row of Bank #a
Ca = at Column Address Ca

Da i = i^{th} word of burst destined to Ba, Ra, Ca

Multiple Accesses to Different Columns in the same Row of a Bank

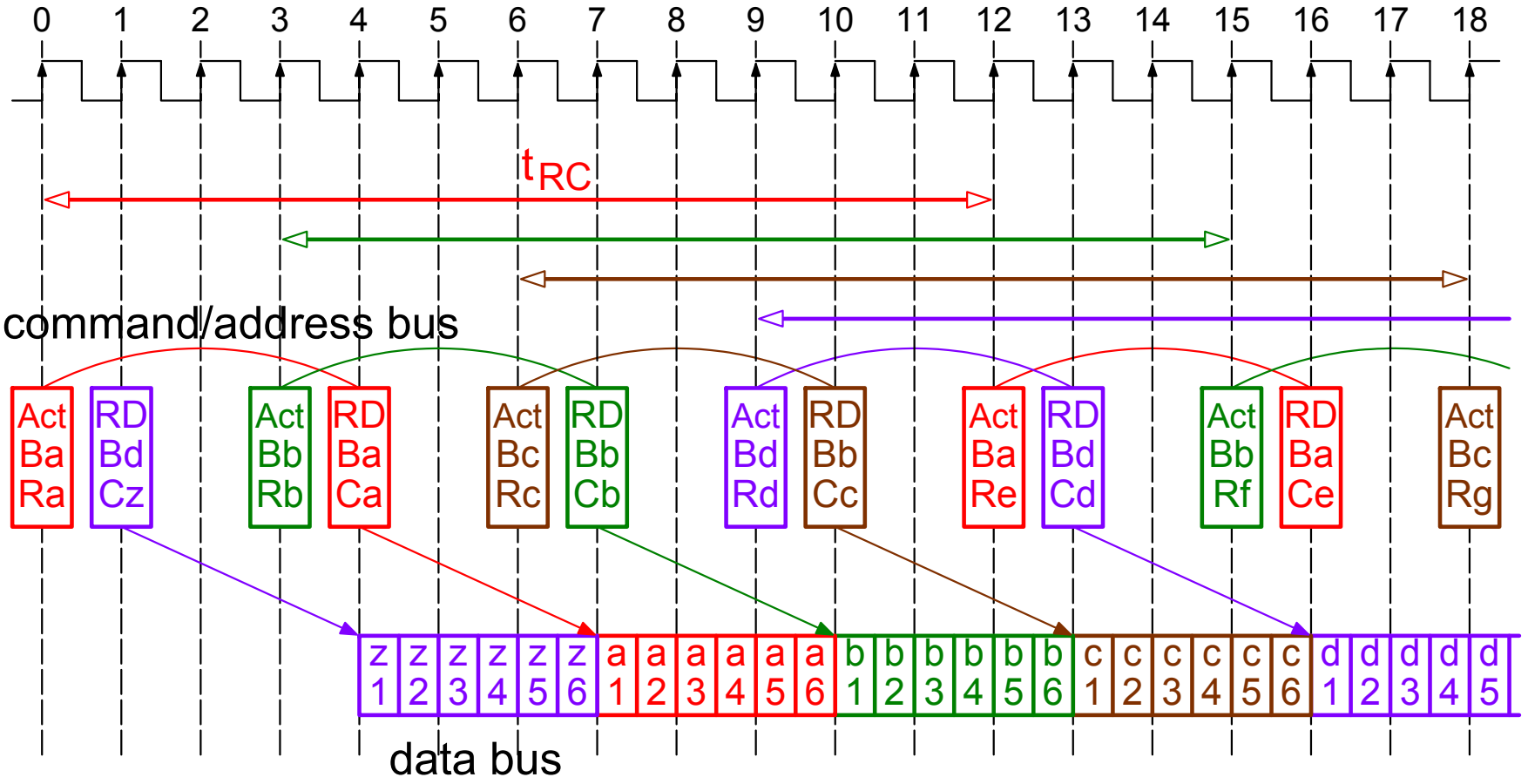


All transactions shown are to the same bank #a, and to the same activated row Ra in that bank.

The transactions shown are:

- Read from Column Ca → a1, a2
- Read from Column Cb → b1, b2
- Write c1, c2 at column Cc
- Write d1, d2 at Column Cd
- Read from column Ce → e1, e2

Multi-Bank Operation: Memory Interleaving



- burst length set to 8; each successive READ command interrupts the preceding burst, resulting in net bursts of 6.