2. Link and Memory Architectures and Technologies

2.1 Links, Thruput/Buffering, Multi-Access Ovrhds

2.2 Memories: On-chip / Off-chip SRAM, DRAM

2.A Appendix: Elastic Buffers for Cross-Clock Commun.

Manolis Katevenis and Giorgos Passas

CS-534 – Univ. of Crete and FORTH, Greece

`www.csd.uoc.gr/~hy534` and `www.ics.forth.gr/~kateveni/534`

2.2 Memories: On-chip / Off-chip SRAM, DRAM

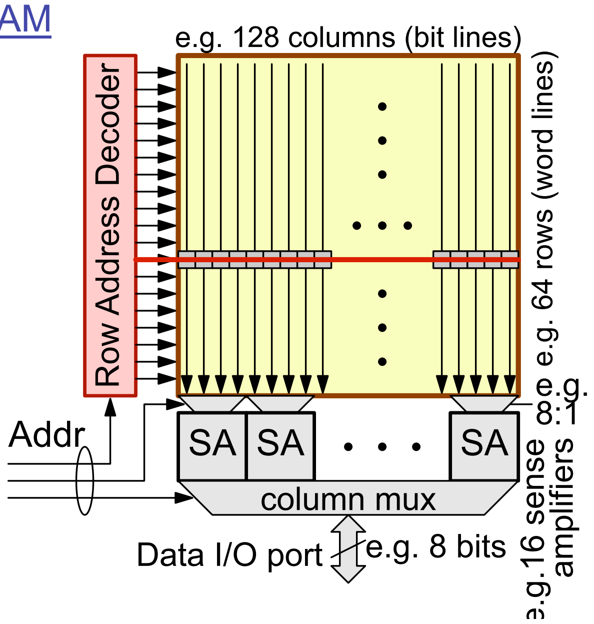
Table of Contents:

- **2.2.1 On-Chip SRAM blocks**
  - Area, Power Consumption, Cycle Time; 1 or 2 ports
  - Power cons. per unit throughput: SRAM, pin transceivers
- **2.2.2 Off-Chip SRAM technologies**
  - Address-Read-Data Pipelining
  - Separate Unidirectional versus Unified Bidirectional Data Lines
- **2.2.3 DRAM Chips and their Pin Interface**
  - Row Access versus Column Access
  - Interleaved accesses to the internal DRAM banks

2.2.1 On-Chip SRAM

Read Cycle Includes:

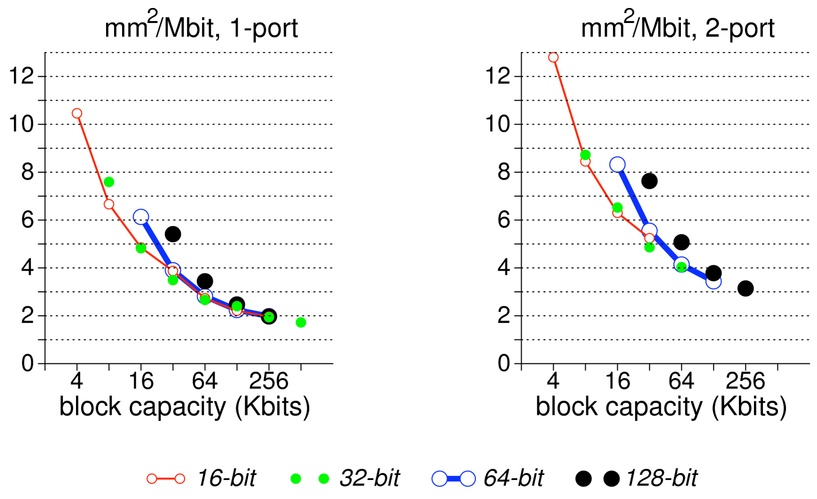
- Precharge bit lines
- Decode row address
- Activate word line
  - faster when narrow
- Discharge bit lines
  - faster when short
- Sense amplifiers
  - don't wait for full discharge before telling the result
- Column multiplexors
  - use column address



Sense Amplifiers: Role, Consequences

- Sense amplifiers significantly speed up read access time
  - sense 0-contents soon after bit-line discharge has started
- Sense amplifiers (SA) are large in size
  - can fit only one SA per 8 columns (sometimes per 4 columns?)
  - analog multiplexors before SA select columns to be read
  - digital multiplexors after SA needed for narrow port widths – they result in large blocks being slower when port is too narrow
- Sense amplifiers consume significant energy when activated
  - only activate the block when read data are actually needed
  - power consumption is proportional to access frequency
  - power consumption is proportional to number of sense amp's (increases with port width, or with bit capacity of SRAM)

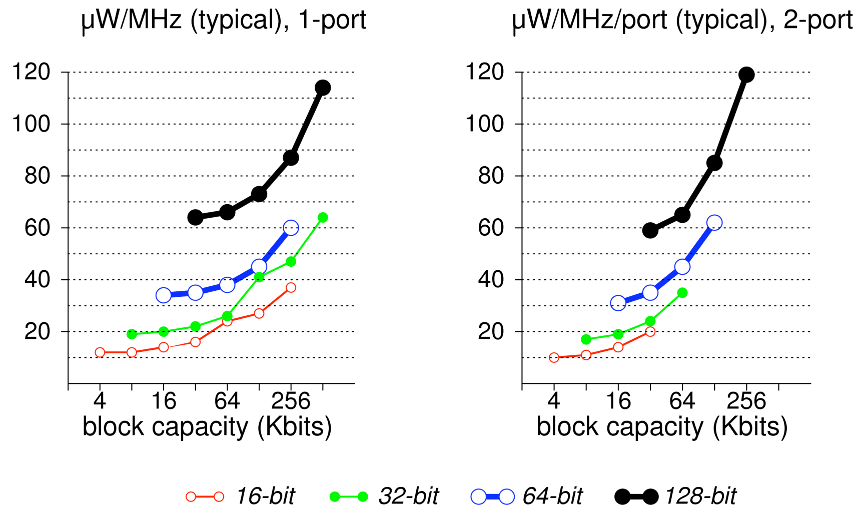
Example on-chip SRAM blocks (90 nm CMOS): **Area**



Area per Megabit: Comments

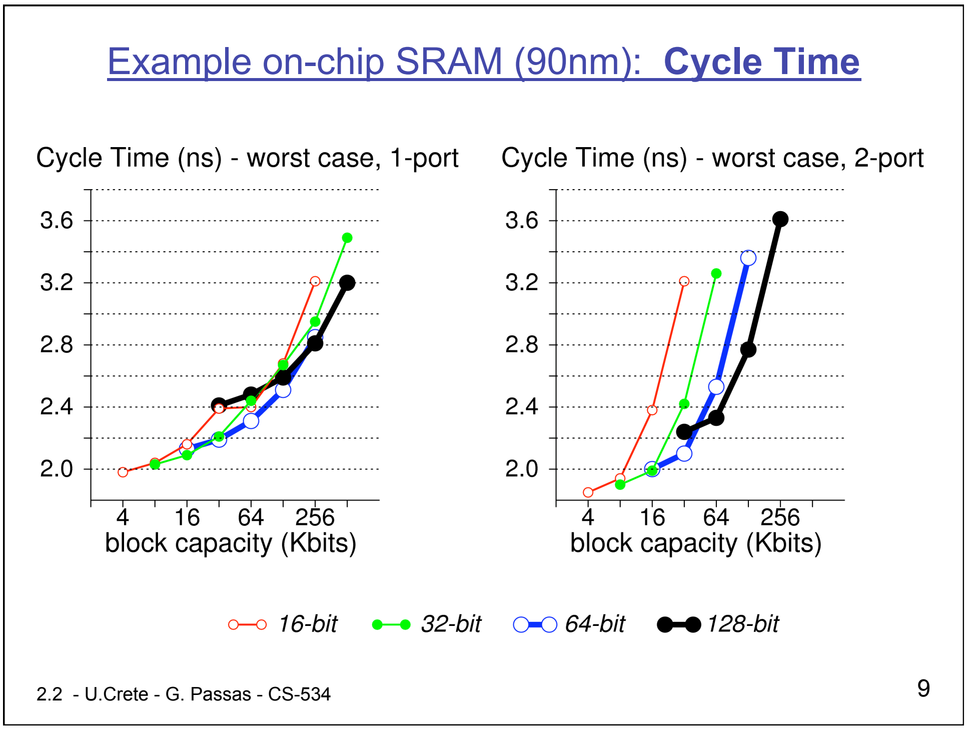
- Slightly old (~2008); values are $(\mu m)^2$/bit = $(mm)^2$/Mbit
- Large blocks are more area-efficient than small ones
  - peripheral overhead (address decoders, column multiplexors, sense amplifiers, power ring) amortized over a larger core
- Port width costs a lot for small blocks
  - more sense amplifiers needed, possibly non-square aspect ratio
  - large blocks need many SA's, for either narrow or wide ports
- Two-port area is about 20 – 60 % more than one-port area
  - core (bit cell) size is the primary reason, hence extra area cost is 20% for smallest blocks and grows to 60% for the largest
- 2-port blocks: *both* ports are rd/wr –*not* one wr- & one rd-port
- Quoted blocks have per-Byte write-enable signals
- Power ring is included in the quoted area figures

Ex. on-chip SRAM (90 nm): **Power Consumption**



Power Consumption (μW/MHz): Comments

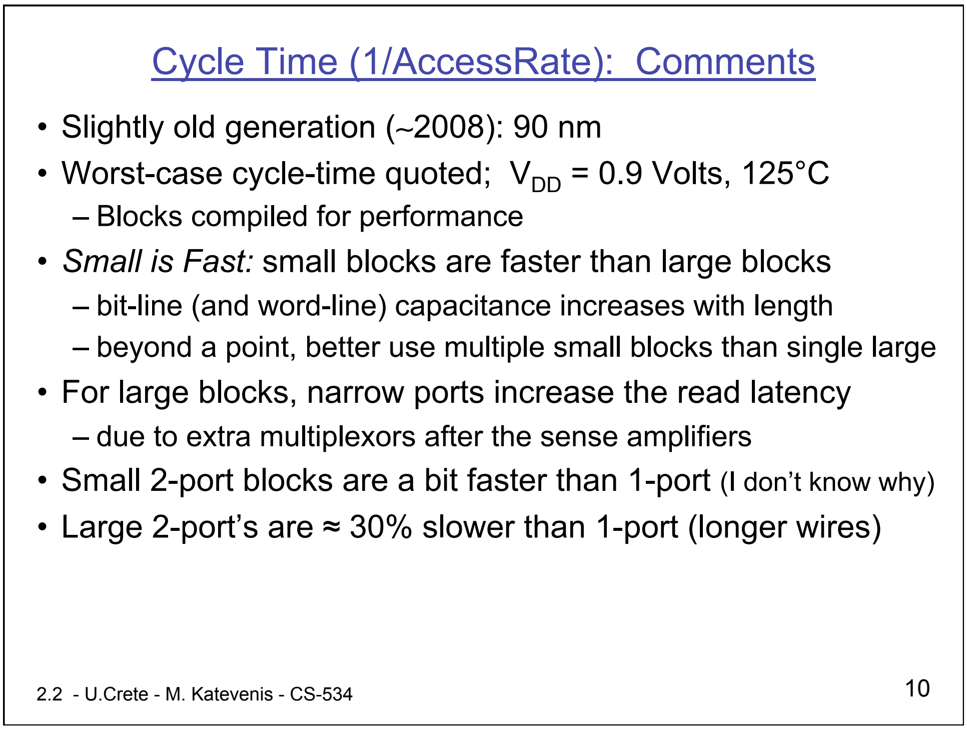
- Slightly old generation (~2008): 90 nm
- Typical-case consumption quoted; V_{DD} = 1.0 Volt, (25°C ?)
  - all cycles active, all address and data bits switching
- Consumption is proportional to access frequency: *μW / MHz*
- Consumption is dominated by port-width, esp. for small blocks
  - actually by the num. of SA's –narrow blocks have more than needed
- Consumption increases with block size due to increasing word-line and bit-line capacitance
  - also increases when size is such that it requires more SA's
- 2-port block consumption is *per-port*
- 2-port *total* consumption ≈ 2x to 3x consumption of 1-port
  - *per-port* consumption is about same for small blocks, but grows to 20 – 50 % more in large 2-port blocks



Example on-chip SRAM (90nm): **Cycle Time**

2.2 - U.Crete - G. Passas - CS-534

9

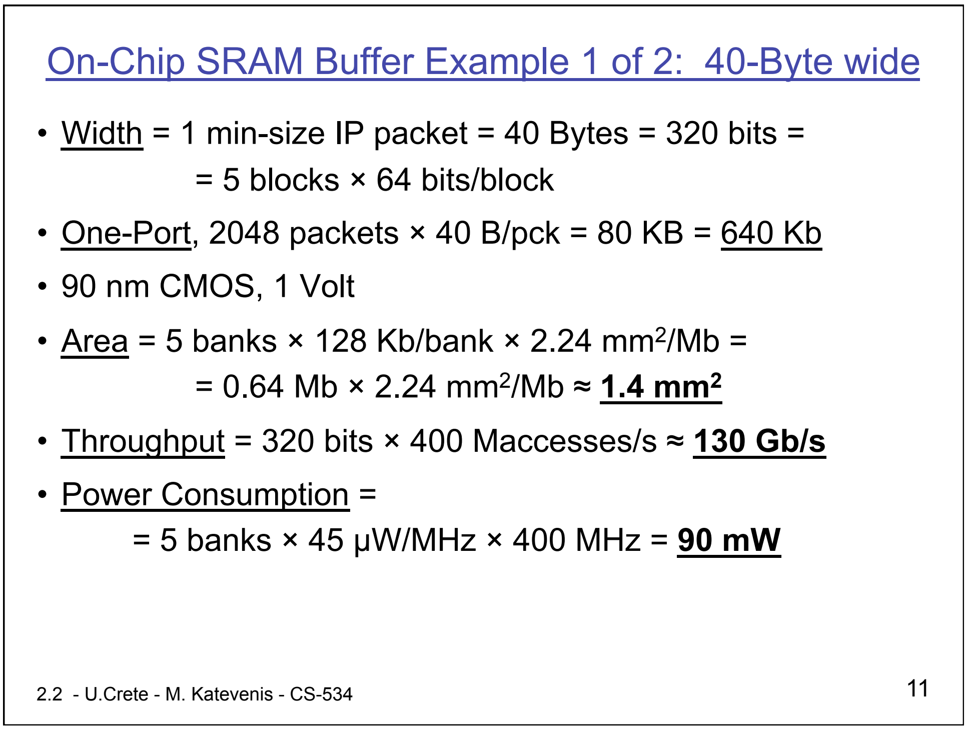


Cycle Time (1/AccessRate): Comments

- Slightly old generation (~2008): 90 nm
- Worst-case cycle-time quoted; V_{DD} = 0.9 Volts, 125°C
  - Blocks compiled for performance
- *Small is Fast:* small blocks are faster than large blocks
  - bit-line (and word-line) capacitance increases with length
  - beyond a point, better use multiple small blocks than single large
- For large blocks, narrow ports increase the read latency
  - due to extra multiplexors after the sense amplifiers
- Small 2-port blocks are a bit faster than 1-port (I don't know why)
- Large 2-port's are ≈ 30% slower than 1-port (longer wires)

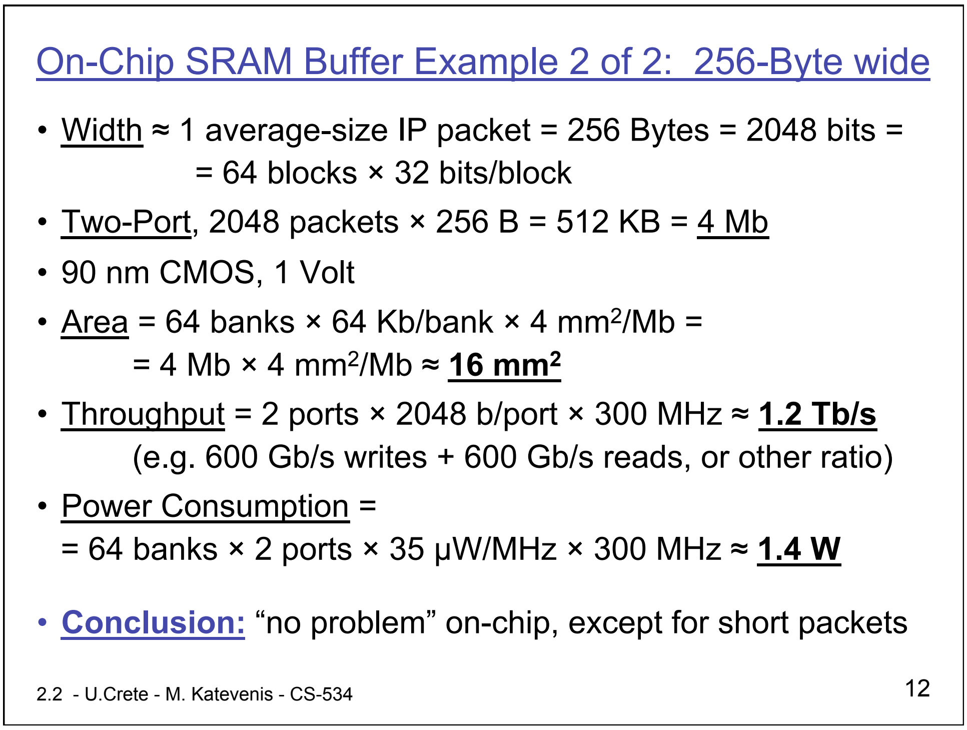
2.2 - U.Crete - M. Katevenis - CS-534

10



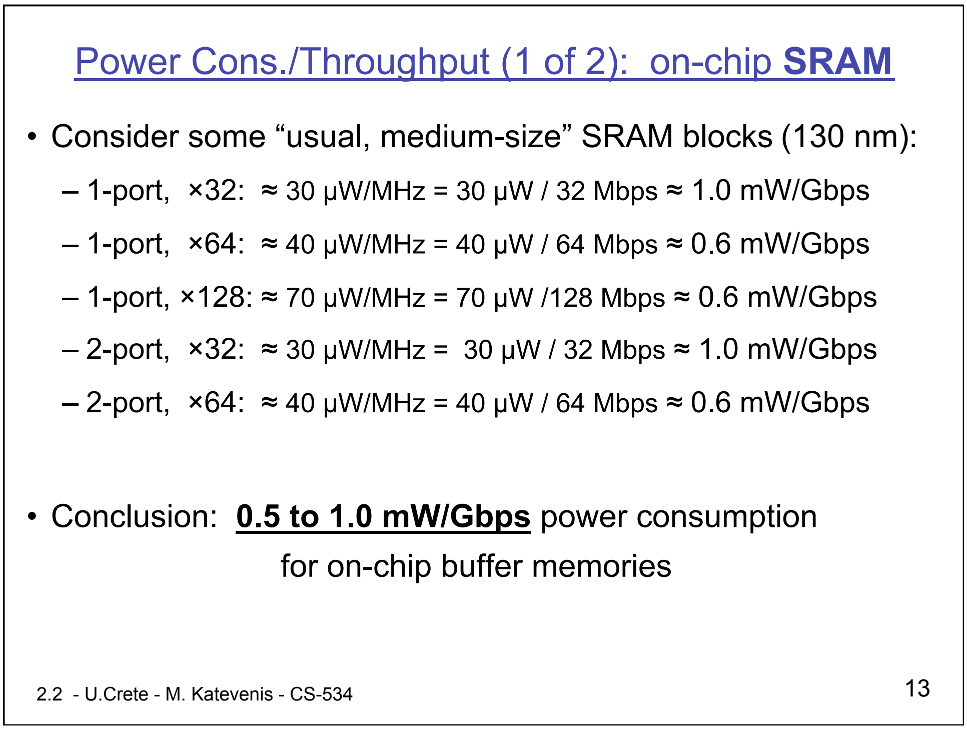
On-Chip SRAM Buffer Example 1 of 2: 40-Byte wide

- Width = 1 min-size IP packet = 40 Bytes = 320 bits =
  = 5 blocks × 64 bits/block
- One-Port, 2048 packets × 40 B/pck = 80 KB = 640 Kb
- 90 nm CMOS, 1 Volt
- Area = 5 banks × 128 Kb/bank × 2.24 mm^2/Mb =
  = 0.64 Mb × 2.24 mm^2/Mb ≈ **1.4 mm^2**
- Throughput = 320 bits × 400 Maccesses/s ≈ **130 Gb/s**
- Power Consumption =
  = 5 banks × 45 μW/MHz × 400 MHz = **90 mW**



On-Chip SRAM Buffer Example 2 of 2: 256-Byte wide

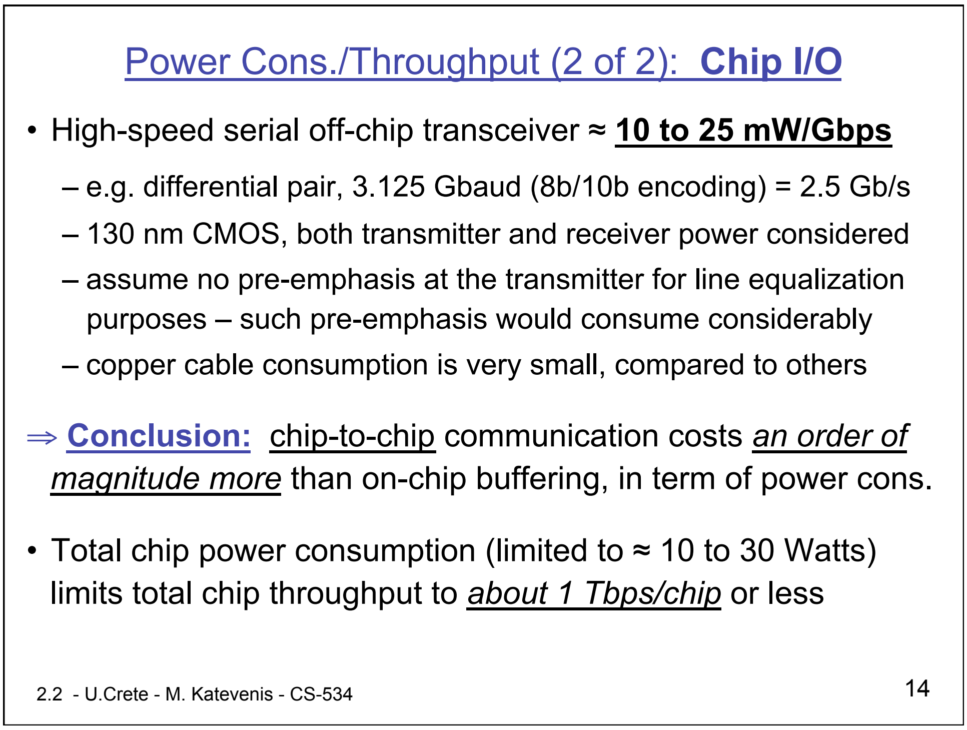
- Width ≈ 1 average-size IP packet = 256 Bytes = 2048 bits =
  = 64 blocks × 32 bits/block
- Two-Port, 2048 packets × 256 B = 512 KB = 4 Mb
- 90 nm CMOS, 1 Volt
- Area = 64 banks × 64 Kb/bank × 4 mm^2/Mb =
  = 4 Mb × 4 mm^2/Mb ≈ **16 mm^2**
- Throughput = 2 ports × 2048 b/port × 300 MHz ≈ **1.2 Tb/s**
  (e.g. 600 Gb/s writes + 600 Gb/s reads, or other ratio)
- Power Consumption =
  = 64 banks × 2 ports × 35 μW/MHz × 300 MHz ≈ **1.4 W**
- **Conclusion:** “no problem” on-chip, except for short packets



Power Cons./Throughput (1 of 2): on-chip **SRAM**

- Consider some “usual, medium-size” SRAM blocks (130 nm):
  - 1-port, ×32: ≈ 30 μW/MHz = 30 μW / 32 Mbps ≈ 1.0 mW/Gbps
  - 1-port, ×64: ≈ 40 μW/MHz = 40 μW / 64 Mbps ≈ 0.6 mW/Gbps
  - 1-port, ×128: ≈ 70 μW/MHz = 70 μW /128 Mbps ≈ 0.6 mW/Gbps
  - 2-port, ×32: ≈ 30 μW/MHz = 30 μW / 32 Mbps ≈ 1.0 mW/Gbps
  - 2-port, ×64: ≈ 40 μW/MHz = 40 μW / 64 Mbps ≈ 0.6 mW/Gbps

- Conclusion: **0.5 to 1.0 mW/Gbps** power consumption
  for on-chip buffer memories



Power Cons./Throughput (2 of 2): **Chip I/O**

- High-speed serial off-chip transceiver ≈ **10 to 25 mW/Gbps**
  - e.g. differential pair, 3.125 Gbaud (8b/10b encoding) = 2.5 Gb/s
  - 130 nm CMOS, both transmitter and receiver power considered
  - assume no pre-emphasis at the transmitter for line equalization purposes – such pre-emphasis would consume considerably
  - copper cable consumption is very small, compared to others

⇒ **Conclusion:** chip-to-chip communication costs *an order of magnitude more* than on-chip buffering, in term of power cons.

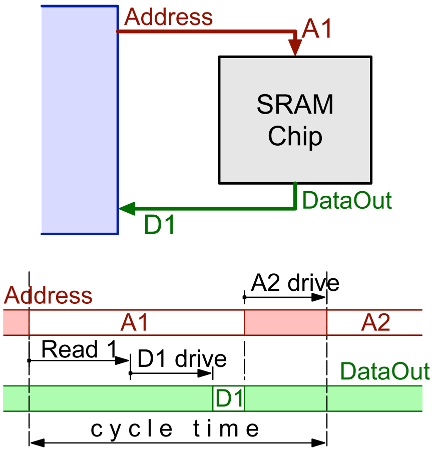
- Total chip power consumption (limited to ≈ 10 to 30 Watts) limits total chip throughput to *about 1 Tbps/chip* or less

2.2.2 Off-Chip SRAM Technologies

- Large on-chip throughput, owing to parallelism of accesses
- Gradual improvements in pin-interface protocols (late 90’s):

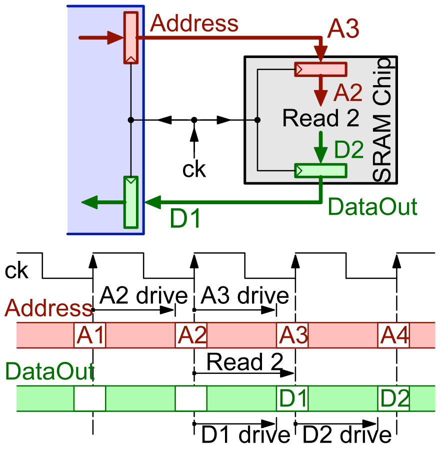
1. Clock-synchronous, pipelined address/data communication
2. Double-Data Rate (DDR) data-pin timing (see §2.1)
3. Source-synchronous clocking
   – clock signal propagating in the same direction as data (or address) signals – normally implies two separate clocks
4. Separate, unidirectional Write-Data and Read-Data buses
   – avoids bus turn-around overhead, but
   – requires 50% writes – 50% reads for full utilization
5. Write-data timing similar to read-data timing
   – first send the address, later send the data, so that address-bus to data-bus time-offset stays fixed for reads & writes

Clock-Synchronous RAM: Pipelined Communication



“Flow Through”: old timing

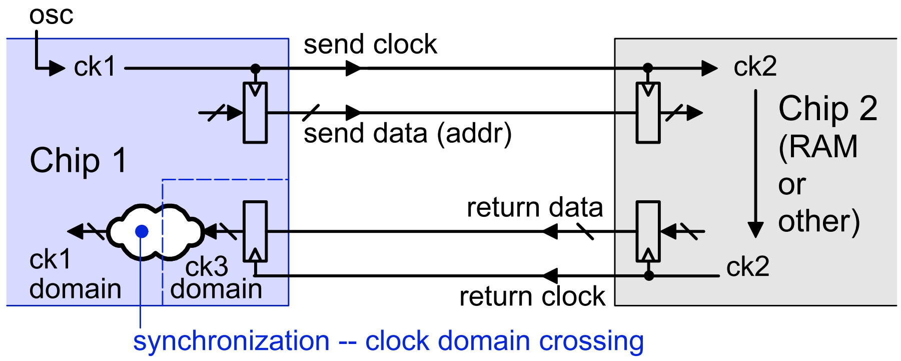
- no overlapping between SRAM operation and communication



“Synchronous” Registered Interface

- pipelined SRAM operation and chip-to-chip communication

Source-Synchronous Data Clocking

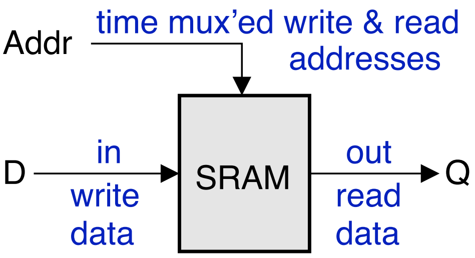


…further increasing the throughput of chip-to-chip communication:

- 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

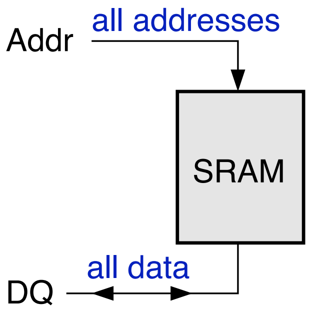
Separate D(in) & Q(out) Paths



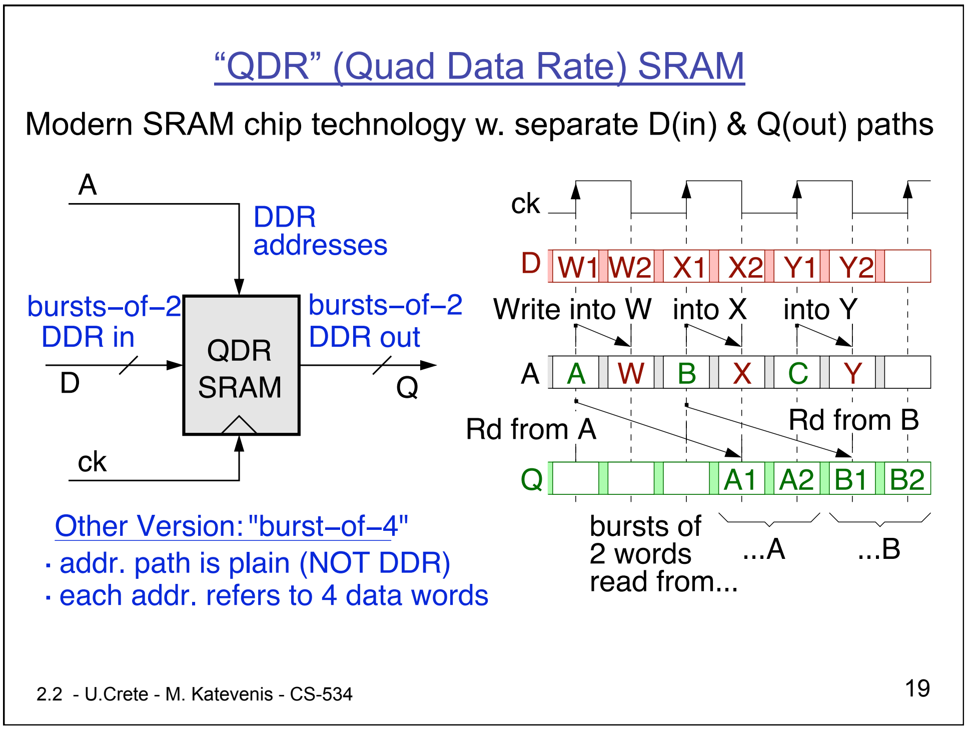
Datapath underutilization when imbalanced (≠ 50 – 50 %) read-write transactions

versus

Shared DQ Data Bus



Bus turn-around overhead: Databus underutilization when frequenctly switching between read and write transactions

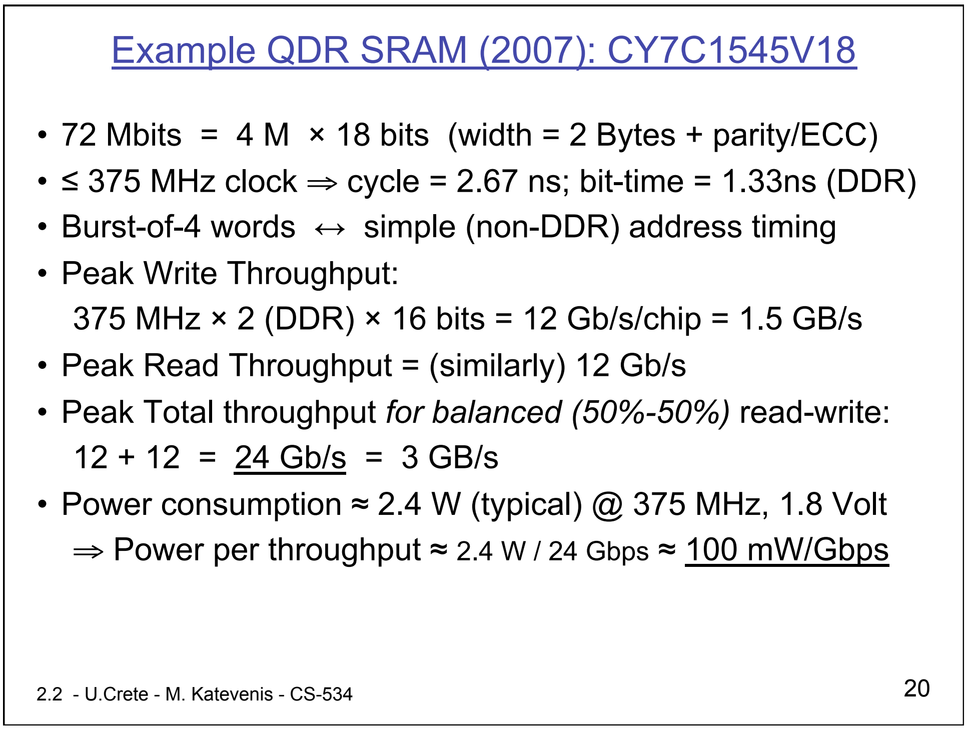


“QDR” (Quad Data Rate) SRAM

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

Other Version: "burst–of–4"

- addr. path is plain (NOT DDR)
- each addr. refers to 4 data words



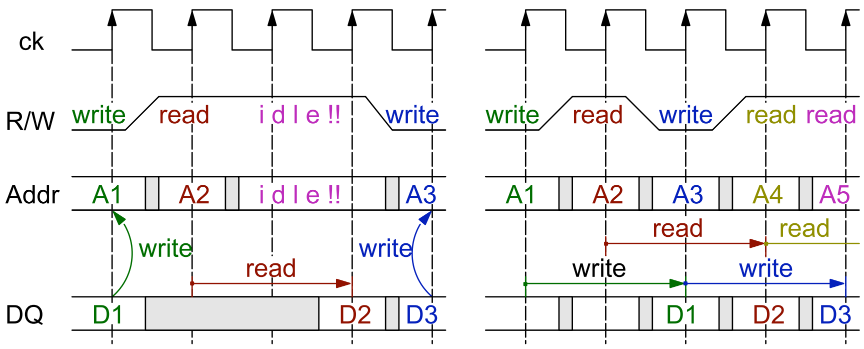
Example QDR SRAM (2007): CY7C1545V18

- 72 Mbits = 4 M × 18 bits (width = 2 Bytes + parity/ECC)
- ≤ 375 MHz clock ⇒ cycle = 2.67 ns; bit-time = 1.33ns (DDR)
- Burst-of-4 words ↔ simple (non-DDR) address timing
- Peak Write Throughput:
  375 MHz × 2 (DDR) × 16 bits = 12 Gb/s/chip = 1.5 GB/s
- Peak Read Throughput = (similarly) 12 Gb/s
- Peak Total throughput *for balanced (50%-50%)* read-write:
  12 + 12 = 24 Gb/s = 3 GB/s
- Power consumption ≈ 2.4 W (typical) @ 375 MHz, 1.8 Volt
  ⇒ Power per throughput ≈ 2.4 W / 24 Gbps ≈ 100 mW/Gbps

Shared "DQ" Data Bus Timing

Naïve Timing

"ZBT" (Zero Bus Turn Around) Timing



Underutilization on every read-to-write transition

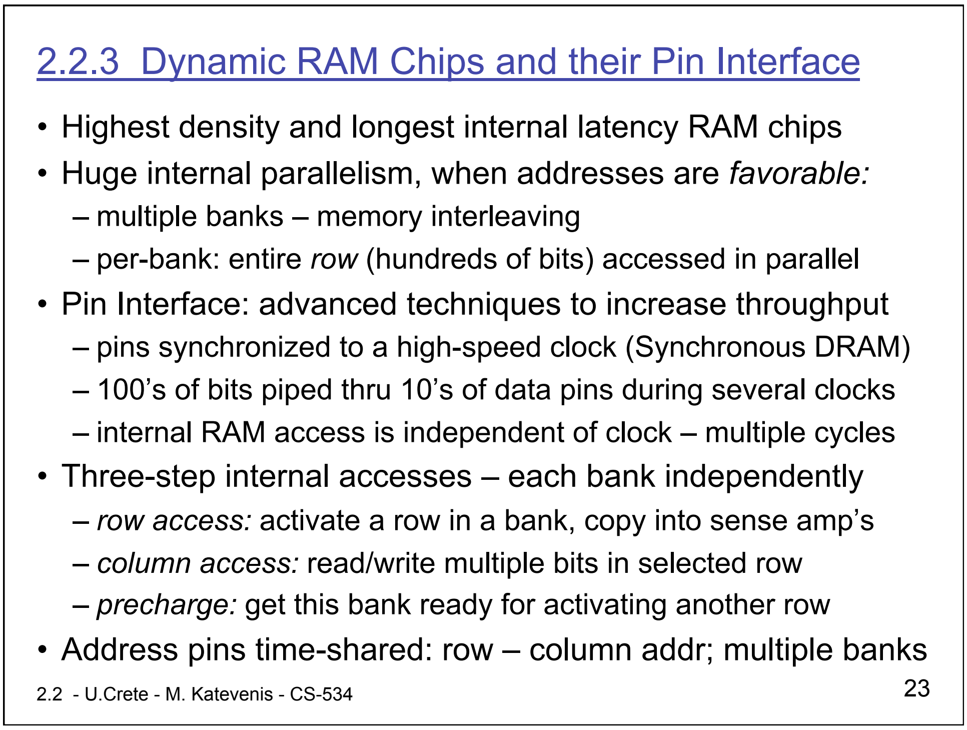
D1 has not yet been written at M[A1] when reading from M[A2] starts… ➔ need to bypass mem. when A2==A1

Example Shared-Bus SRAM (2007): CY7C1550V18

- 72 Mbits = 2 M × 36 bits (width = 4 Bytes + parity/ECC)
- ≤ 375 MHz clock ⇒ cycle = 2.67 ns; bit-time = 1.33ns (DDR)
- Peak Throughput = 375 MHz × 2 (DDR) × 32 bits = 24 Gb/s
- "NoBL" (No Bus Latency) = "ZBT" (Zero Bus Turn-Around, ala Micron)
- Although NoBL/ZBT, one clock cycle is lost every time the bus direction changes from read to write (bus turn-around)

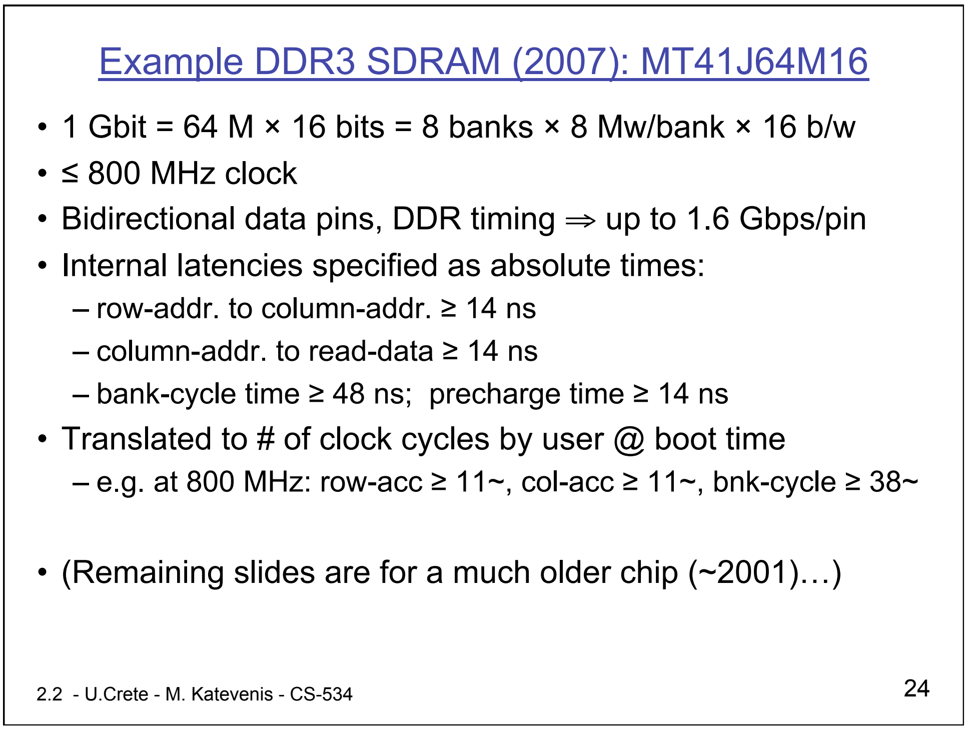
  ⇒ throughput with alternating read/writes ≈
  ≈ 2/3 × peak throughput ≈ 16 Gb/s
- Power consumption ≈ 2.4 W (typical) @ 375 MHz, 1.8 Volts

  ⇒ Power per throughput ≈ 2.4 W / 24 Gbps ≈ 100 mW/Gbps



2.2.3 Dynamic RAM Chips and their Pin Interface

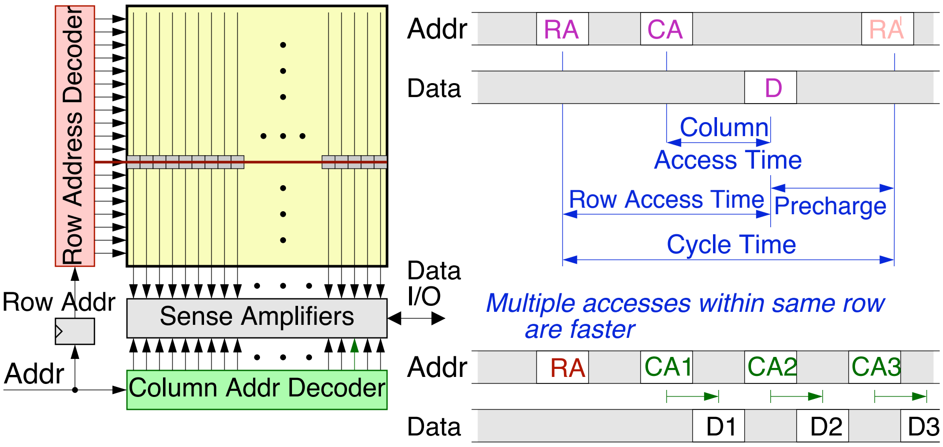
- Highest density and longest internal latency RAM chips
- Huge internal parallelism, when addresses are *favorable:*
  - multiple banks – memory interleaving
  - per-bank: entire *row* (hundreds of bits) accessed in parallel
- Pin Interface: advanced techniques to increase throughput
  - pins synchronized to a high-speed clock (Synchronous DRAM)
  - 100's of bits piped thru 10's of data pins during several clocks
  - internal RAM access is independent of clock – multiple cycles
- Three-step internal accesses – each bank independently
  - *row access:* activate a row in a bank, copy into sense amp's
  - *column access:* read/write multiple bits in selected row
  - *precharge:* get this bank ready for activating another row
- Address pins time-shared: row – column addr; multiple banks



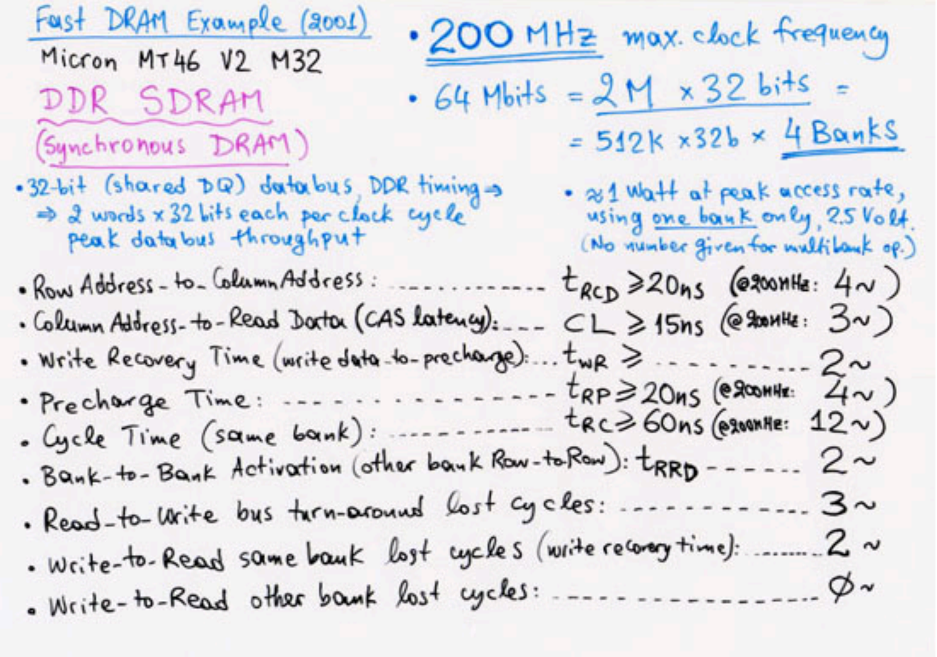
Example DDR3 SDRAM (2007): MT41J64M16

- 1 Gbit = 64 M × 16 bits = 8 banks × 8 Mw/bank × 16 b/w
- ≤ 800 MHz clock
- Bidirectional data pins, DDR timing ⇒ up to 1.6 Gbps/pin
- Internal latencies specified as absolute times:
  - row-addr. to column-addr. ≥ 14 ns
  - column-addr. to read-data ≥ 14 ns
  - bank-cycle time ≥ 48 ns; precharge time ≥ 14 ns
- Translated to # of clock cycles by user @ boot time
  - e.g. at 800 MHz: row-acc ≥ 11~, col-acc ≥ 11~, bnk-cycle ≥ 38~
- (Remaining slides are for a much older chip (~2001)…)

DRAM Basics:
Row Address, Column Address, Precharge



Multiple accesses within same row are faster



Fast DRAM Example (2001)

Micron MT46 V2 M32

DDR SDRAM

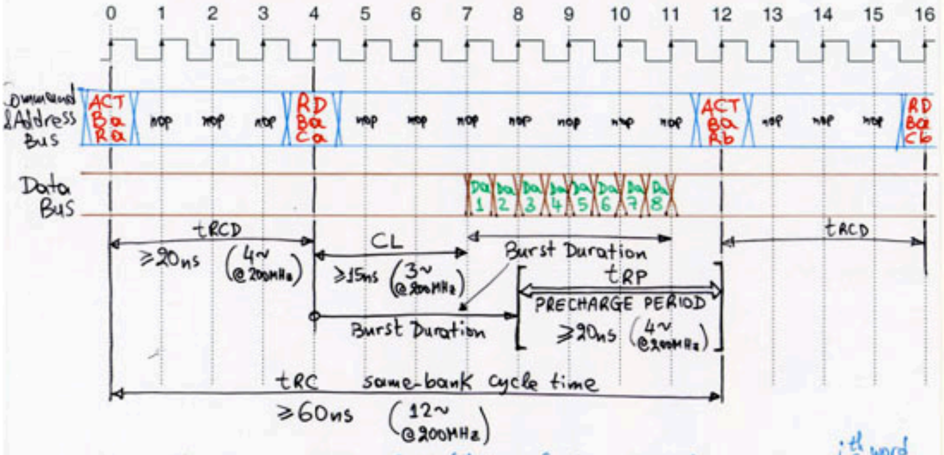
(Synchronous DRAM)

- **200 MHz** max. clock frequency
- 64 Mbits = 2M × 32 bits = 512K × 32b × 4 Banks

- 32-bit (shared DQ) databus, DDR timing → ⇒ 2 words × 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: .... $t_{RCD} \geq 20$ns (@200MHz: 4~)
- Column Address-to-Read Data (CAS latency): ... CL ≥ 15ns (@200MHz: 3~)
- Write Recovery Time (write data-to-precharge): ... $t_{WR} \geq$ .... 2~
- Precharge Time: .... $t_{RP} \geq 20$ns (@200MHz: 4~)
- Cycle Time (same bank): .... $t_{RC} \geq 60$ns (@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: .... Ø~

Single-Bank Read Access

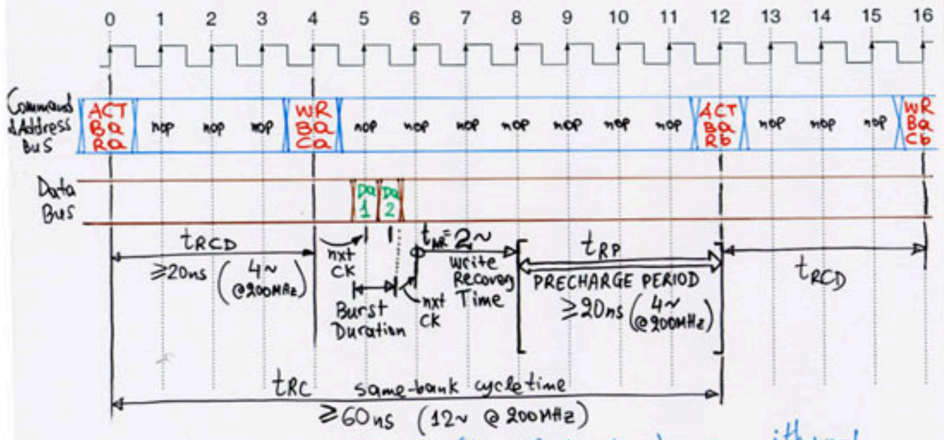


ACT = Activate
Ba = Bank #a
Ra = Row # Ra Address

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

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

Single-Bank Write Access

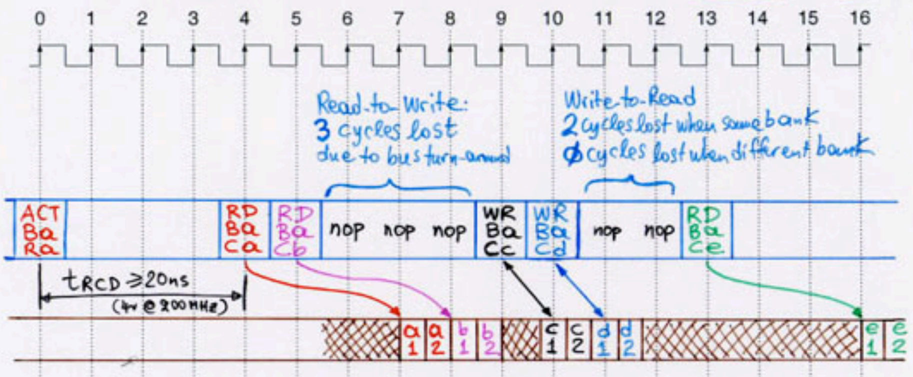


ACT = Activate
Ba = Bank #a
Ra = Row Address Ra

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

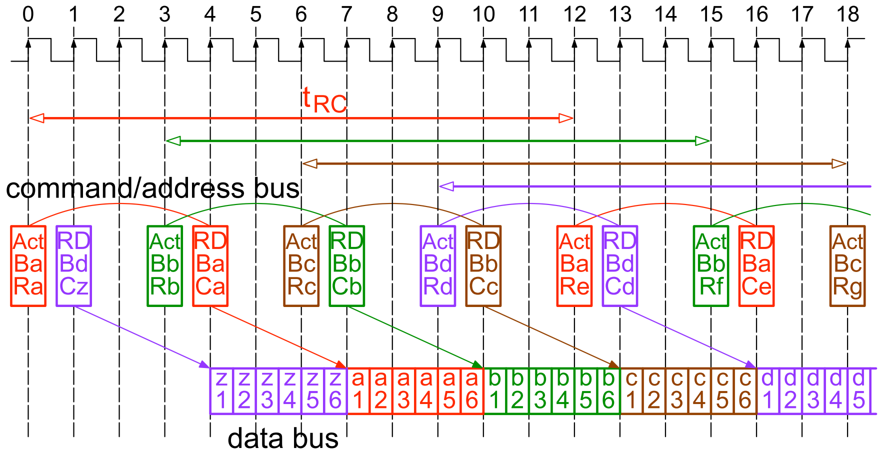


All transactions shown are to the same bank #a, and to the same activated row Ra in that bank.

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.