

An introduction to SDRAM and memory controllers

Benny Akesson



TU / **e**

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Presentation Outline

Introduction to SDRAM

Basic SDRAM operation

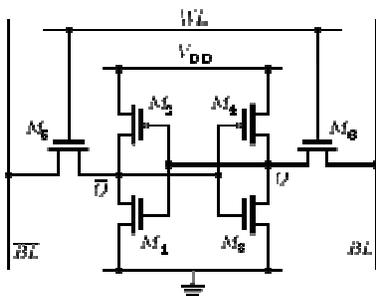
Memory efficiency

SDRAM controller architecture

Conclusions

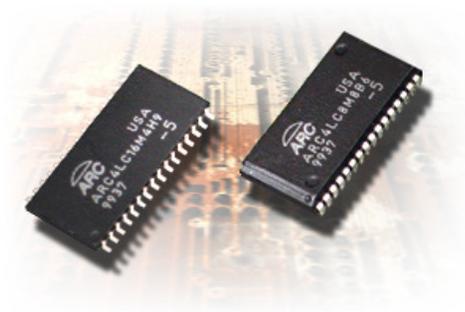
Static RAM (SRAM)

- ▶ SRAM is on-chip memory
- ▶ Found in higher levels of the memory hierarchy
 - Commonly used for caches and scratchpads
- ▶ Either local to processor or centralized
 - Local memory has very short access time
 - Centralized shared memories have intermediate access time
- ▶ An SRAM cell consists of six transistors
 - Limits memory to a few megabytes, or even smaller



Dynamic RAM (DRAM)

- ▶ DRAM was patented in 1968 by Robert Dennard at IBM
- ▶ Significantly cheaper than SRAM
 - DRAM cell has 1 transistor and 1 capacitor vs. 6 transistors for SRAM
 - A bit is represented by a high or low charge on the capacitor
 - Charge dissipates due to leakage – hence the term dynamic RAM
 - Capacity of up to a gigabyte per chip
- ▶ DRAM is (shared) off-chip memory
 - Long access time compared to SRAM
 - Off-chip pins are expensive in terms of area and power
 - SDRAM bandwidth is **scarce** and must be **efficiently** utilized
- ▶ Found in lower levels of memory hierarchy
 - Used as remote high-volume storage



The DRAM evolution

- ▶ Evolution of the DRAM design in the past 15 years
 - A clock signal was added making the design synchronous (SDRAM)
 - The data bus transfers data on both rising and falling edge of the clock (DDR SDRAM)
 - Second and third generation of DDR memory (DDR2/DDR3) scales to higher clock frequencies (up to 800 MHz)
 - DDR4 currently being standardized by JEDEC
 - Special branches of DDR memories for graphic cards (GDDR) and for low power systems (LPDDR)

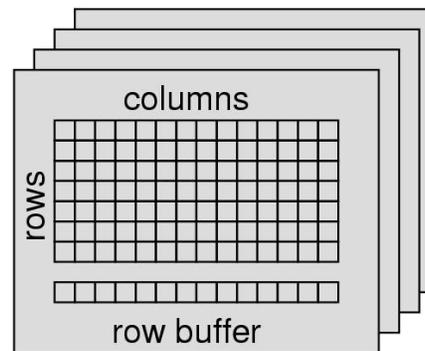


SDRAM Architecture

- ▶ The SDRAM architecture is organized in **banks**, **rows** and **columns**
 - A row buffer stores a currently active (open) row
- ▶ The memory interface has a **command bus**, **address bus**, and a **data bus**
 - Busses shared between all banks to reduce the number of off-chip pins
 - A bank is essentially is an independent memory, but with shared I/O

Typical values DDR2/DDR3:

4 or 8 banks
8K – 65K rows / bank
1K – 2K columns / row
4, 8, 16 bits / column
200-800 MHz
32 MB – 1 GB density



Example memory:

16-bit DDR2-400B 64 MB
4 banks
8K rows / bank
1024 columns / row
16 bits / column

800 MB/s peak
bandwidth



Presentation Outline

Introduction to SDRAM

Basic SDRAM operation

Memory efficiency

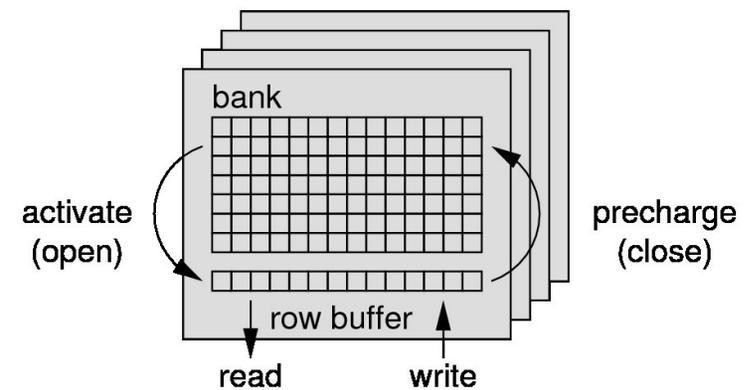
SDRAM controller architecture

Conclusions

Basic SDRAM Operation

- ▶ Requested row is **activated** and copied into the row buffer of the bank
- ▶ Read bursts and/or write **bursts are issued to the active row**
 - Programmed **burst length** (BL) of 4 or 8 words
- ▶ Row is **precharged** and stored back into the memory array

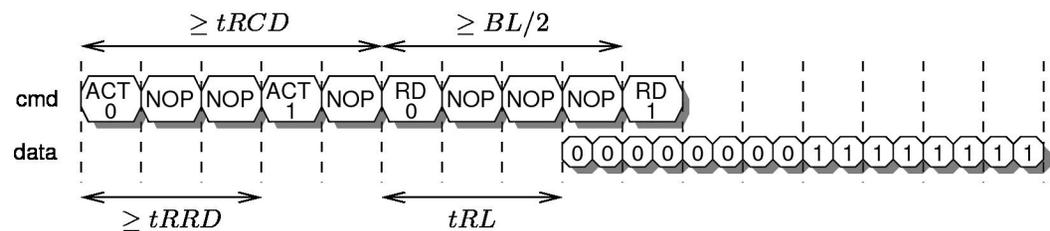
Command	Abbr	Description
Activate	ACT	Activate a row in a particular bank
Read	RD	Initiate a read burst to an active row
Write	WR	Initiate a write burst to an active row
Precharge	PRE	Close a row in a particular bank
Refresh	REF	Start a refresh operation
No operation	NOP	Ignores all inputs



Timing Constraints

- ▶ **Timing constraints** determine which commands can be scheduled
 - More than 20 constraints, some are inter-dependent
 - Limits the efficiency of memory accesses
 - Wait for precharge, activate and read/write commands before data on bus
 - Timing constraints get increasingly severe for faster memories
 - The physical design of the memory core has not changed much
 - Constant in nanoseconds constant, but clock period gets shorter

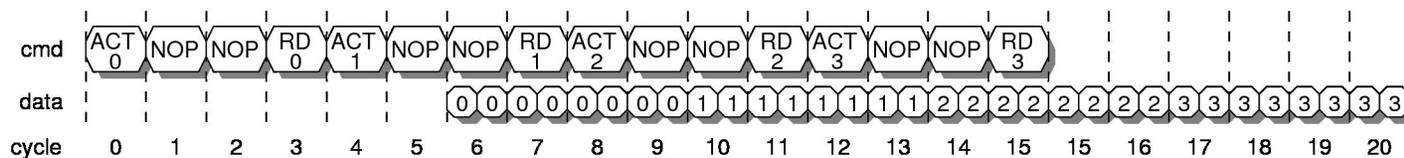
Parameter	Abbr.	Cycles
ACT to RD/WR	tRCD	3
ACT to ACT (diff. banks)	tRRD	2
ACT to ACT (same bank)	tRAS	12
Read latency	tRL	3
RD to RD	-	BL/2



Pipelined SDRAM access

- ▶ Multiple banks provide parallelism
 - SDRAM has separate data and command buses
 - Commands to different banks are **pipelined**
 - Activate, precharge and transfer data in parallel (**bank preparation**)
 - Increases efficiency

- ▶ Figure shows pipelined memory accesses with burst length 8



Presentation Outline

Introduction to SDRAM

Basic SDRAM operation

Memory efficiency

SDRAM controller architecture

Conclusions

Memory Efficiency

- ▶ **Memory efficiency** is the fraction of clock cycles with data transfer
 - Defines the **exchange rate** between **peak bandwidth** and **net bandwidth**
 - Net bandwidth is the actual useful bandwidth after considering overhead

- ▶ Five categories of memory efficiency for SDRAM:
 - Refresh efficiency
 - Read/write efficiency
 - Bank efficiency
 - Command efficiency
 - Data efficiency

- ▶ Memory efficiency is the product of these five categories

Refresh Efficiency

- ▶ SDRAM need to be refreshed regularly to retain data
 - DRAM cell contains leaking capacitor
 - Refresh command must be issued every $7.8 \mu\text{s}$ for DDR2/DDR3 SDRAM
 - All banks must be precharged
 - Data cannot be transferred during refresh

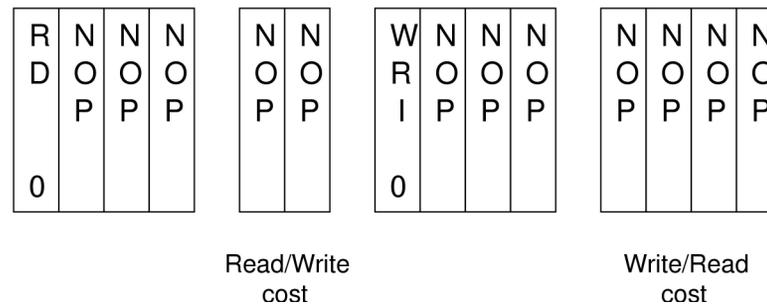
- ▶ Refresh efficiency is largely **independent of traffic**
 - Depends on density of the memory device (generally 95 – 99%)



Read / Write Efficiency

- ▶ Cycles are lost when switching direction of the data bus
 - Extra NOPs must be inserted between read and write commands

- ▶ Read/write efficiency **depends on traffic**
 - Determined by frequency of read/write switches
 - Switching too often has a significant impact on memory efficiency
 - Switching after every burst of 8 words gives 57% r/w efficiency with DDR2-400



Bank Efficiency

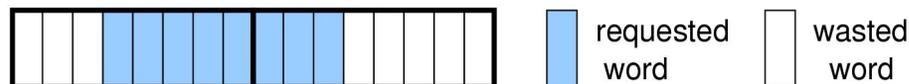
- ▶ Bank conflict when a read or write targets an inactive row (row miss)
 - Significantly impacts memory efficiency
 - Requires precharge followed by activate
 - Less than 40% bank efficiency if always row miss in same bank
- ▶ Bank efficiency **depends on traffic**
 - Determined by address of request and memory map

Command Efficiency

- ▶ Command bus uses single data rate
 - Congested if precharge and activate is required simultaneously
 - One command has to be delayed – may delay data on bus
- ▶ Command efficiency **depends on traffic**
 - Small bursts reduce command efficiency
 - Potentially more activate and precharge commands issued
 - Generally quite high (95-100%)

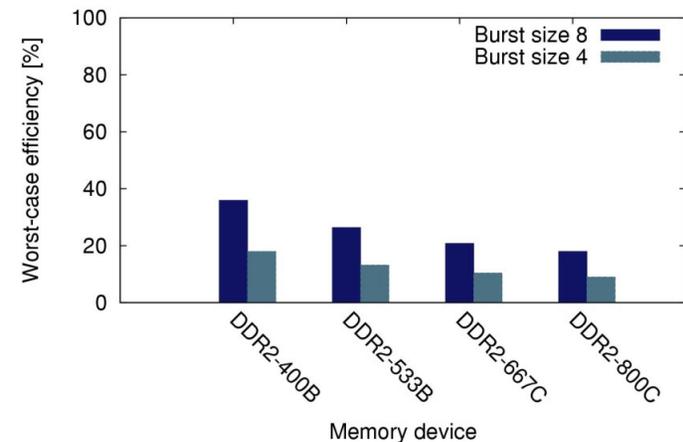
Data Efficiency

- ▶ A memory burst can access segments of the programmed burst size.
 - Minimum access granularity
 - Burst length 8 words is 16 B with 16-bit memory and 64 B with 64-bit memory
- ▶ If data is poorly aligned an extra segment have to be transferred
 - Cycles are lost when transferring unrequested data
- ▶ Data efficiency **depends on the application**
 - Smaller requests and bigger burst length reduce data efficiency



Conclusions on Memory Efficiency

- ▶ Memory efficiency is highly dependent on traffic
- ▶ Worst-case efficiency is very low
 - Every burst targets different rows in the same bank
 - Read/write switch after every burst
- ▶ Results in
 - Less than 40% efficiency for all DDR2 memories
 - Efficiency drops as memories become faster (DDR3)
- ▶ Conclusion
 - Worst-case efficiency must be prevented!



Presentation Outline

Introduction to SDRAM

Basic SDRAM operation

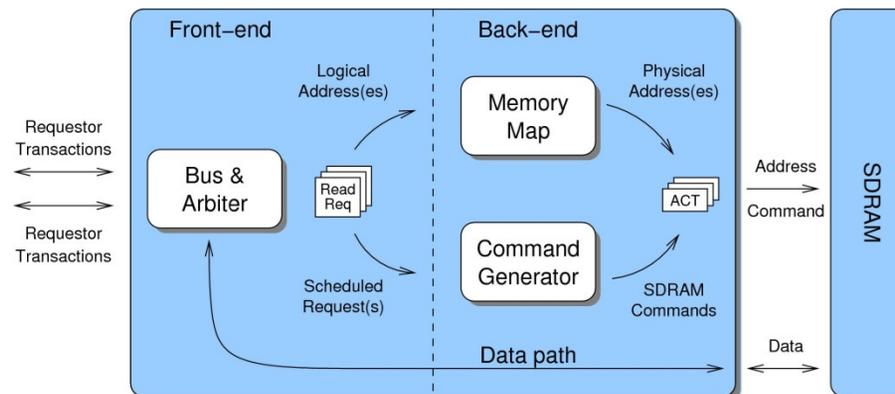
Memory efficiency

SDRAM controller architecture

Conclusions

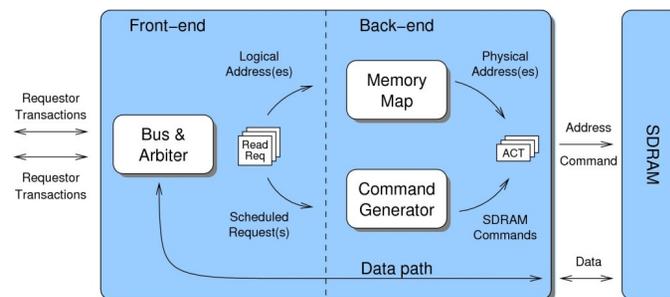
A general memory controller architecture

- ▶ A general controller architecture consists of two parts
- ▶ The **front-end**
 - buffers requests and responses per requestor
 - schedules one (or more) requests for memory access
 - is independent of the memory type
- ▶ The **back-end**
 - translates scheduled request(s) into SDRAM command sequence
 - is dependent on the memory type



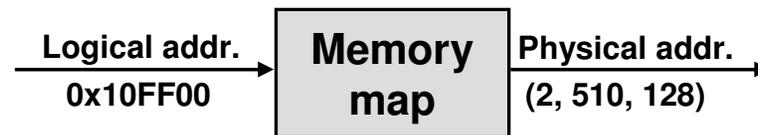
Front-end arbitration

- ▶ Front-end provides **buffering** and **arbitration**
- ▶ Arbiter can schedule requests in many different ways
 - Priorities common to give low-latency access to critical requestors
 - E.g. stalling processor waiting for a cache line
 - Important to prevent starvation of low priority requestors
 - Common to schedule fairly in case of multiple processors
 - Next request may be scheduled before previous is finished
 - Gives more options to command generator in back-end
- ▶ Scheduled requests are sent to the back-end for memory access



Back-end

- ▶ Back-end contains a **memory map** and a **command generator**
- ▶ Memory map decodes logical address to physical address
 - Physical address is (bank, row, column)
 - Can be done in different ways – choice affects efficiency



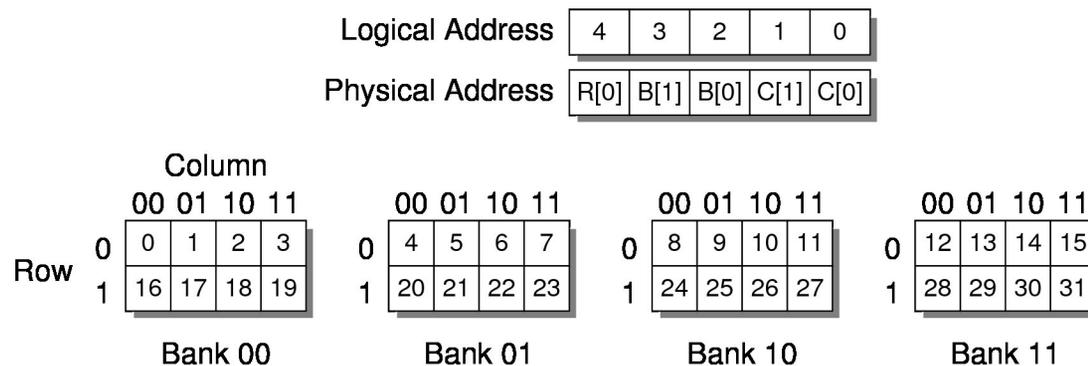
- ▶ Command generator schedules commands for the target memory
 - Customized for a particular memory generation
 - Programmable to handle different timing constraints



Continuous memory map

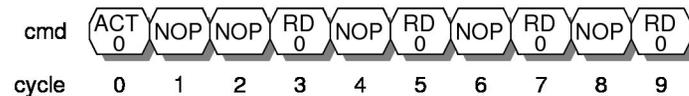
- ▶ The memory map decodes a memory address into (bank, row, column)
 - Decoding is done by slicing the bits in the logical address

- ▶ Continuous memory map
 - Map sequential address to columns in row
 - Switch bank when all columns in row are visited
 - Switch row when all banks are visited

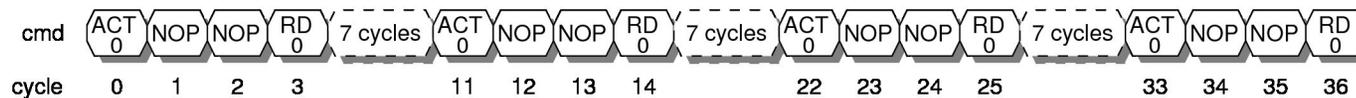


Continuous memory map

- ▶ Continuous memory map **very sensitive to locality**
- ▶ Advantage:
 - Very efficient in best case
 - No bank conflicts when reading sequential addresses
 - 10 cycles to issue four read commands with burst length 4 words

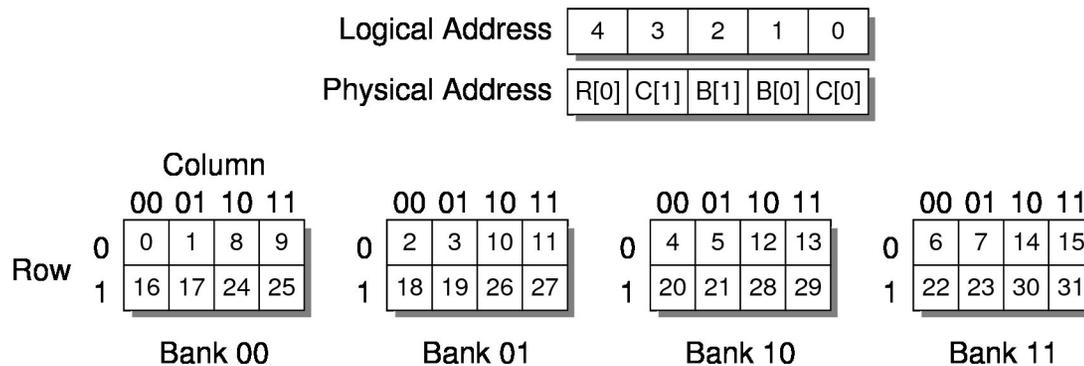


- ▶ Disadvantage:
 - Very inefficient if requesting different rows in same bank
 - 37 cycles to issue the four read commands



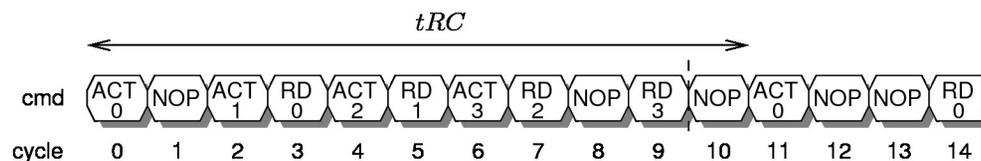
Interleaving memory map

- ▶ Interleaving memory map
 - Maps bursts to different banks in interleaving fashion
 - Active row in a bank is not changed until all columns are visited



Interleaving memory map

- ▶ Interleaving memory map is **largely insensitive to locality**
- ▶ Advantage:
 - Makes extensive use of bank parallelism to hide overhead
 - Average case and worst case is almost the same
 - Takes 10 or 11 cycles for the four read commands depending on locality
 - Compare to 10 or 37 for continuous memory map
- ▶ Disadvantages:
 - Requires bursts to all banks to be efficient
 - Solved if requests are large, such as 64 B
 - Issues many activate and precharges which increases power consumption



Command generator

- ▶ Generates and schedules commands for scheduled requests
 - May work with both requests and commands
- ▶ Many ways to determine which **request** to process
 - Increase bank efficiency
 - Prefer requests targeting open rows
 - Increase read/write efficiency
 - Prefer read after read and write after write
 - Reduce stall cycles of processor
 - Always prefer reads, since reads are blocking and writes often posted

Command generator

- ▶ Generate SDRAM commands without violating timing constraints
 - Often use bank controller to determine valid commands to a bank
- ▶ Many possible policies to determine which **command** to schedule
 - Precharge policies
 - Close rows as soon as possible to activate new one faster
 - Keep rows open as long as possible to benefit from locality
 - Command priorities
 - Read and write commands have high priority, as they put data on the bus
 - Precharge and activate commands have lower priorities
 - Algorithms often try to put data on the bus as soon as possible
 - Microsoft proposes a self-learning memory controller that uses reinforcement-learning to do long-term planning

Presentation Outline

Introduction to SDRAM

Basic SDRAM operation

Memory efficiency

SDRAM controller architecture

Conclusions

Conclusions

- ▶ SDRAM is used as shared off-chip high-volume storage
 - Cheaper but slower than SRAM
- ▶ The efficiency of SDRAM is **highly variable** and depends on
 - Refresh efficiency, bank efficiency, read/write efficiency, command efficiency, and data efficiency
- ▶ Controller tries to **minimize latency** and **maximize efficiency**
 - Low-latency for critical requestors using priorities
 - Fairness among multiple processors
 - High efficiency by reordering requests to fit with memory state
- ▶ Memory map impacts efficiency
 - Continuous memory map good if small requests and good locality
 - Interleaving memory map good if large requests and poor locality

Questions?

k.b.akesson@tue.nl