COPYRIGHT

DISCLAIMER

This content is provided without warranty or representation of any kind. The use of the content is entirely at your own risk and Stellenbosch University (SU) will have no liability directly or indirectly as a result of this content.

The content must not be assumed to provide complete coverage of the particular study material. Content may be removed or changed without notice.

The video is of a recording with very limited post-recording editing. The video is intended for use only by SU students enrolled in the particular module.

forward together · saam vorentoe · masiye phambili

Computer Systems / Rekenaarstelsels 245

Lecture 28

External Memory/
Eksterne Geheue

Dr Rensu Theart & Dr Lourens Visagie

External Memory Motivation
Redes vir Eksterne Geheue

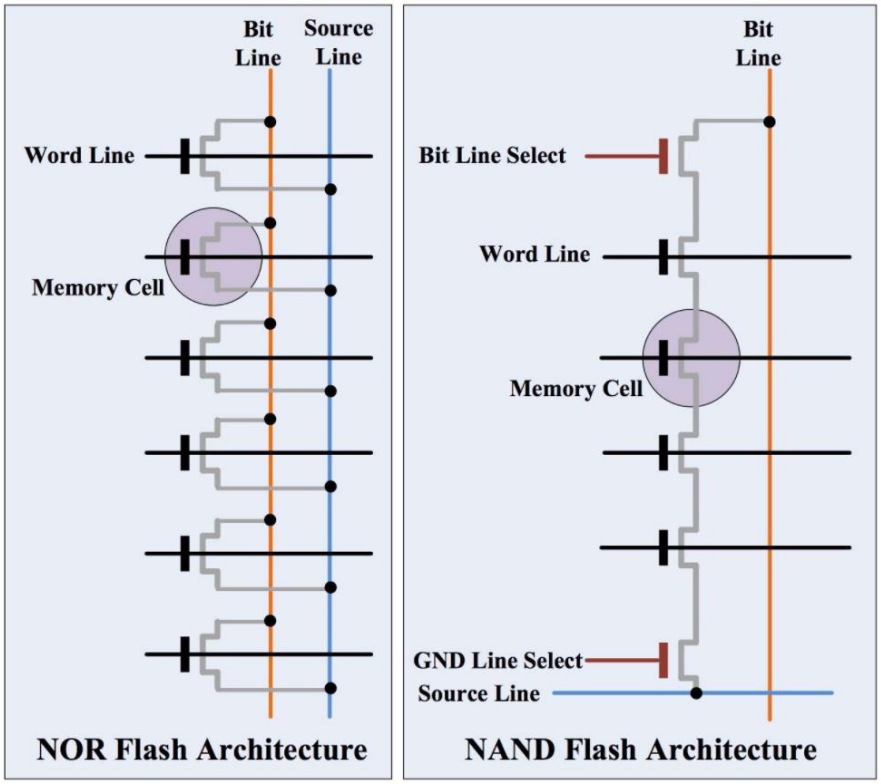
- Most of the time, the available memory in the microcontroller is sufficient for the application (SRAM + Flash)
  - Choose MCU variant with enough capability
- Sometimes, the application demands more memory, or memory of a different kind
  - External data files that won't fit in flash memory
  - Data logging
  - SDRAM for LCD display frame buffer (double-buffering)
  - Storing camera images/video
  - Program or non-volatile data that must be erasable and writeable separate from main program
  - FPGA connects to microcontroller through memory-mapped interface

Types of External Memory
Tipes Eksterne Geheue

- Removable storages, like SD cards, USB, make use of serial interfaces. We've already covered that
- Permanent external memories
  - NAND Flash
  - NOR Flash
  - SDRAM
- These differ by their implementation, and also how they are interfaced with
- The microcontroller requires an external memory controller peripheral for this (STM microcontroller have FSMC – Flexible Static Memory Controller, SiLabs MCUs have EBI – External Bus Interface, etc.)
- Mostly available on larger devices – parallel memory interface increased package pin count

Flash Memory Implementations
Flash Geheue Implementerings

- Flash memories store information in memory cells made from floating gate transistors
- Flash memory comes in NAND and NOR flavours
- In NOR Flash,
  - one end of each memory cell is connected to the source line and the other end directly to a bit line resembling a NOR Gate
  - NOR Flash memory cells are connected in parallel, thus "random access" is possible: A bit of information can be extracted using parallel address lines
  - Fast read speeds but lower density (compared to NAND)
- In NAND Flash, several memory cells (typically eight cells) are connected in series similar to a NAND gate
  - Random access is not possible – interface to NAND device is though an I/O mapped or indirect interface
  - Read speed is slower, but has faster write and erase speeds (compared to NOR flash)
  - NAND flash devices are manufactured with imperfections – require ECC (Error Correction Code) functionality

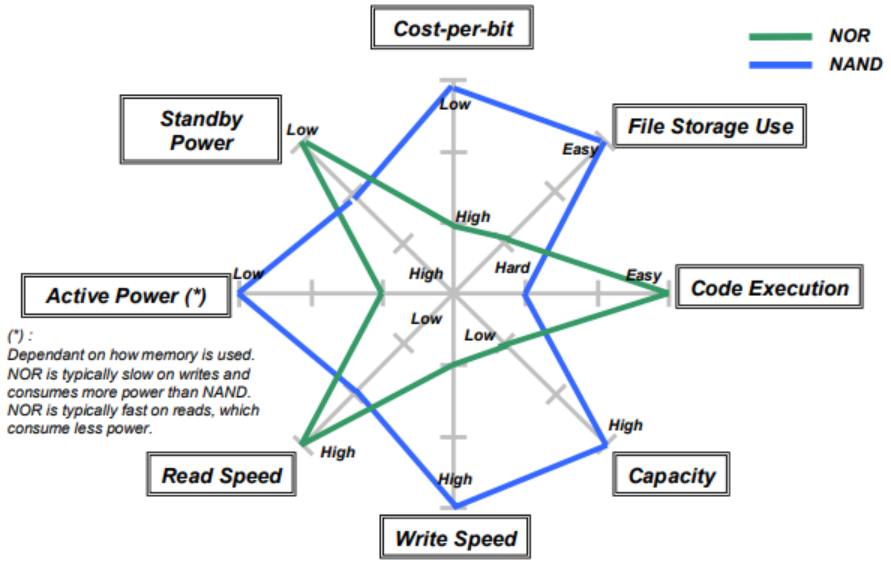


https://www.embedded.com/flash-101-nand-flash-vs-nor-flash/

Flash Memory Comparison

Flash Geheue Vergelyking

Fig. 1 Comparison of NOR and NAND Flash



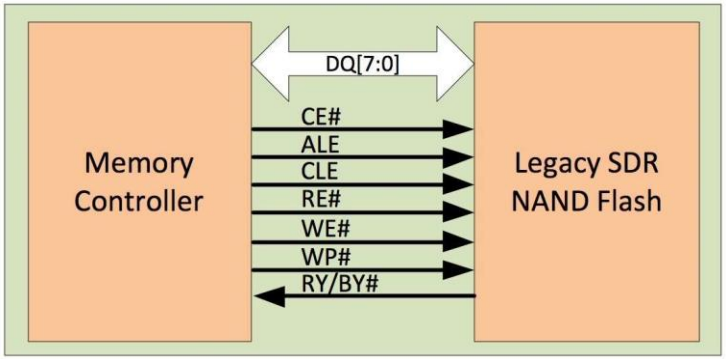
Toshiba NAND vs. NOR Flash Memory Technology Overview

- In general,
  - NOR flash is used for program code
  - NAND flash is used for mass storage
- MCU external memory interfaces usually allows both types of interfaces

NAND Flash Interface

NAND Flash Koppelvlak

- Legacy NAND interface
  - Asynchronous (no clock signal) – data transactions are governed by control signals
  - 8-bit bi-directional data bus
- Newer NAND interfaces (ONFI NAND and Toggle NAND) add more signals but with increased throughput

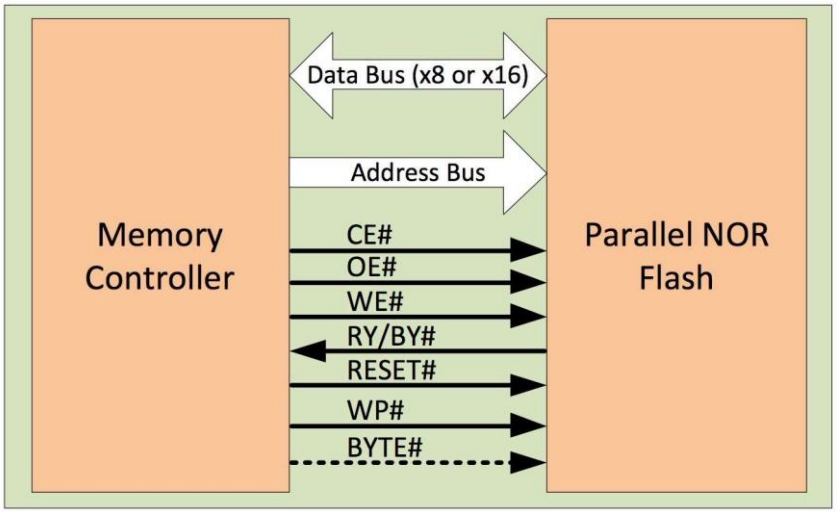


Signal		Function
DQ [7:0]	Data bus	Bidirectional, 8-bit data bus
CE#	Chip Enable	Input Signal, logic low selects the device for data transfer with the host memory controller.
ALE	Address Latch Enable	Input Signal, one of the control signal used by host controller to indicate the type of bus cycle (command, address or data)
CLE	Command Latch Enable	Input Signal, one of the control signal used by host controller to indicate the type of bus cycle (command, address or data)
RE#	Read Enable	Input Signal, control signal for read data transfer. Data is transferred at the rising edge of RE#
WE#	Write Enable	Input Signal, control signal for write data transfer. Data, command and address are latched at the rising edge of WE#
WP#	Write Protect	Input Signal, disables Flash array program and erase operations
RY/BY#	Ready/ Busy	Output Signal, indicates whether the device is executing any operation or ready for next operation.

NOR Flash Interface

NOR Flash Koppelvlak

- NOR Flash Interface is the same as asynchronous SRAM
- Parallel data and address busses
- Data bus width can be variable (8, 16 or 32-bit)
- Number of address lines, N:
- 2^N = memory capacity (bits) / number of data bits
- Random Access: MCU can select which word of data is wants to read/write by setting appropriate address bits
- Sometimes lower address bits are multiplexed on the data bus – need an additional signal (Latch Enable or Address Valid)



Signal		Function
CE#	Chip Enable	Input Signal, logic low selects the device for data transfer with the host memory controller.
OE#	Output Enable	Input Signal, controls whether outputs signals are actively driven or in high impedance
WE#	Write Enable	Input Signal, controls the direction of data transfer between host and device.
RY/BY#	Ready/Busy	Output Signal, indicates whether the device is executing any operation or ready for next operation.
RESET#		Input Signal, hardware reset, causes the device to reset control logic to its standby state.
WP#	Write Protect	Input Signal, disables program and erase functions for the protected sector of the device.
BYTE#		Input Signal, indicates the data bus width for devices with 8-bit & 16-bit data bus support

NOR Flash/SRAM Interface

NOR Flash/SRAM Koppelvlak

- Memory access is asynchronous (clock is not passed on to memory
- Reading from memory:
  - MCU outputs address
  - Chip Enable = enabled
  - Output Enable = enable
  - Write Enable = disabled
  - Memory drives data bus (outputs requested
- Writing to memory
  - MCU outputs address
  - Chip enable = enabled
  - Output enable = disabled
  - Write enabled = enabled
  - MCU outputs data on data bus

Figure 35. Mode1 read access waveforms

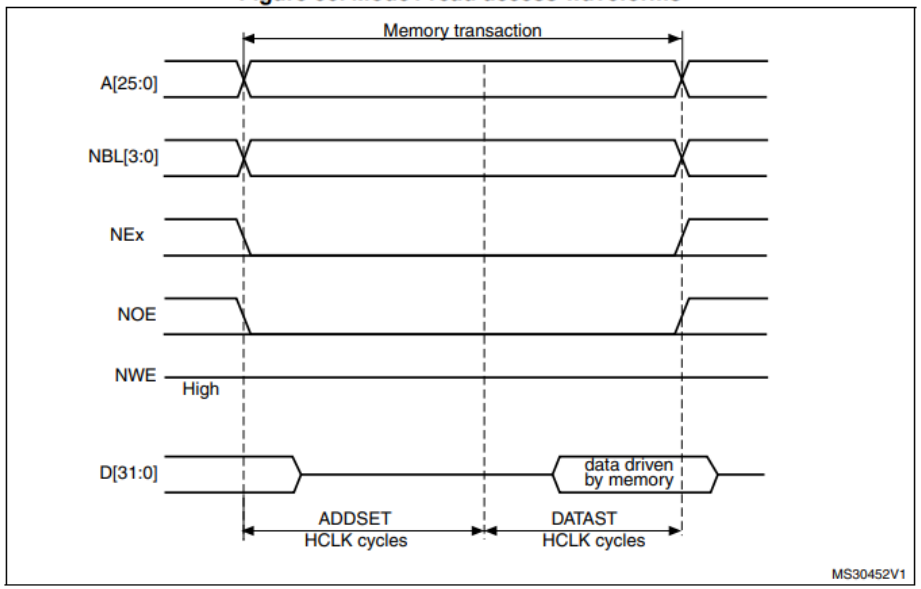
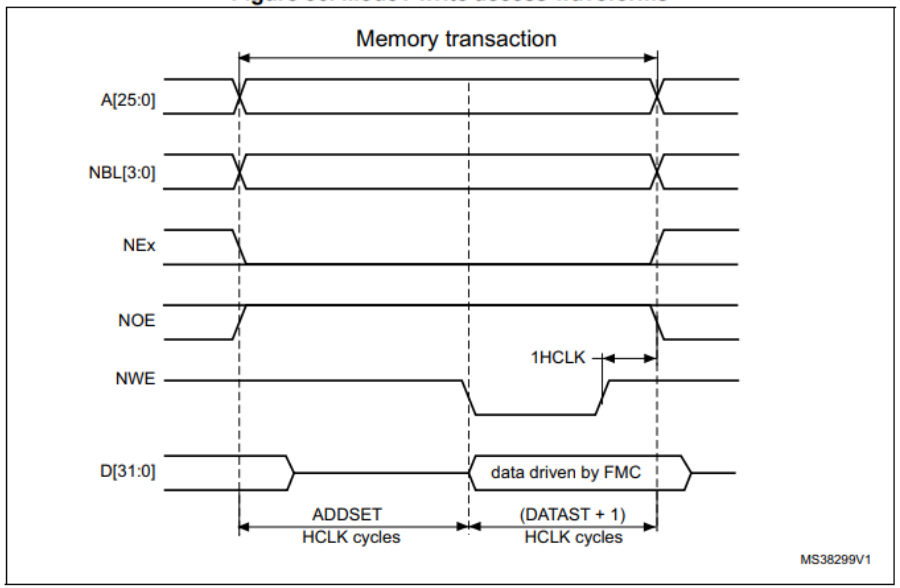
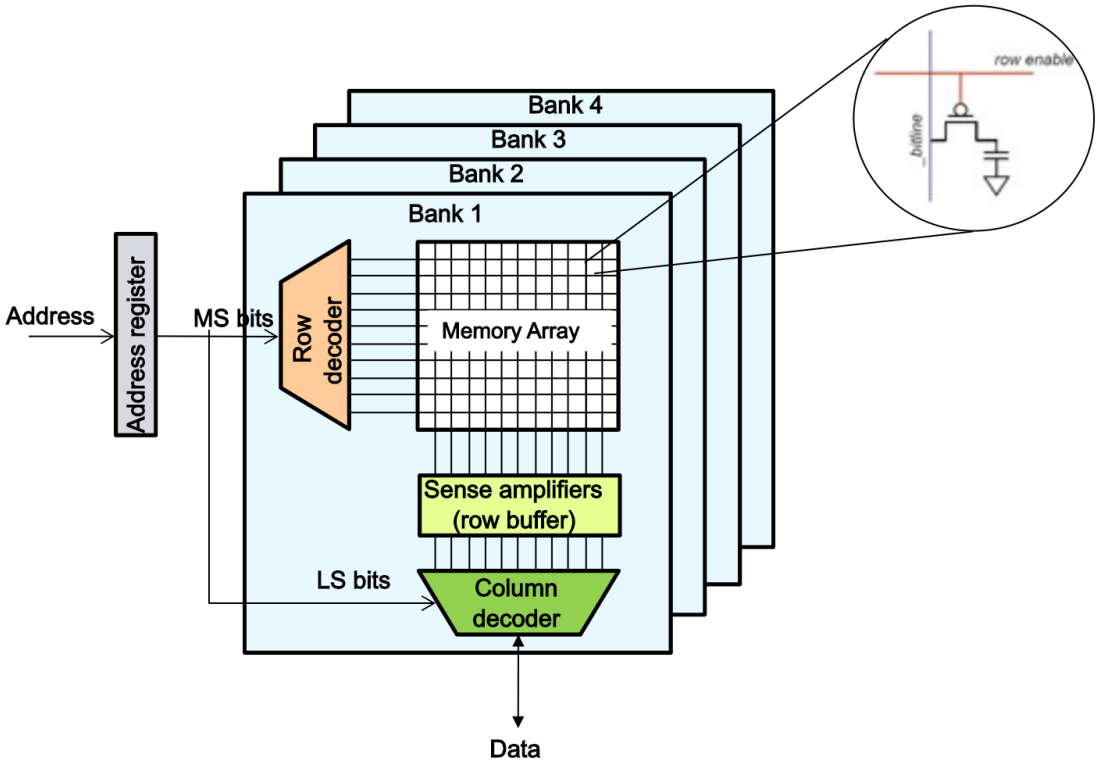


Figure 36. Mode1 write access waveforms



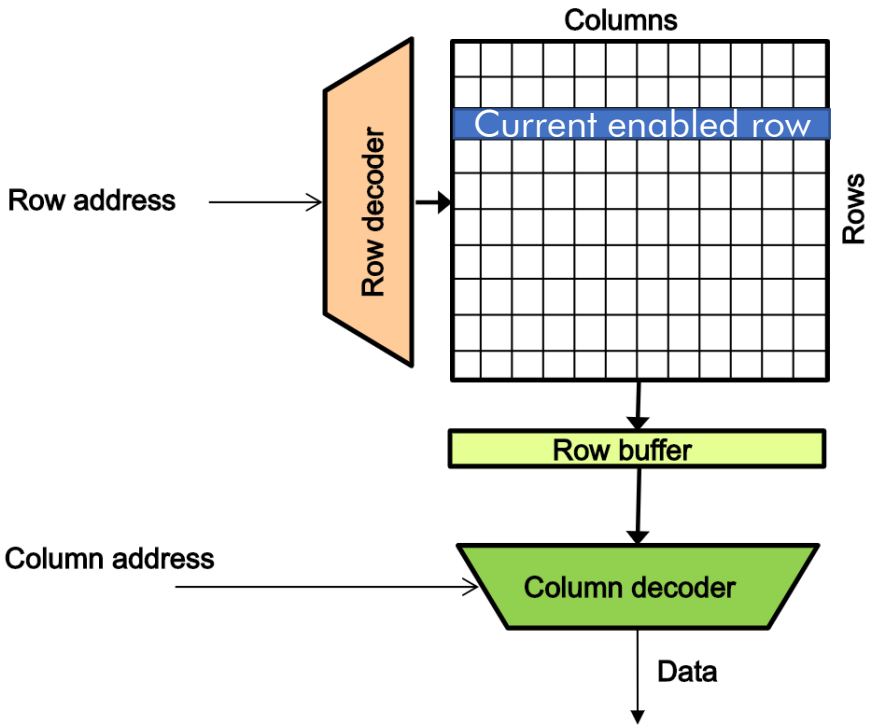
SDRAM Implementation
SDRAM Implementering

- Information is stored as charge on a capacitor
- Memory cells are arranged in a matrix, with rows and columns
- Sometimes multiple banks, each consisting of a matrix of memory cells



SDRAM Implementation
SDRAM Implementering

- To reduce pin-count, row and column selection is performed on the same address lines
- Control signals are used to differentiate which is it – row or column address
  - RAS = Row Address Strobe
  - CAS = Column Address Strobe
- Once a row is enabled it is possible to address multiple columns on the same row
  - This reduces latency because the row address does not have to be re-sent and set-up
  - Enabling a new row takes time (read cycle time)

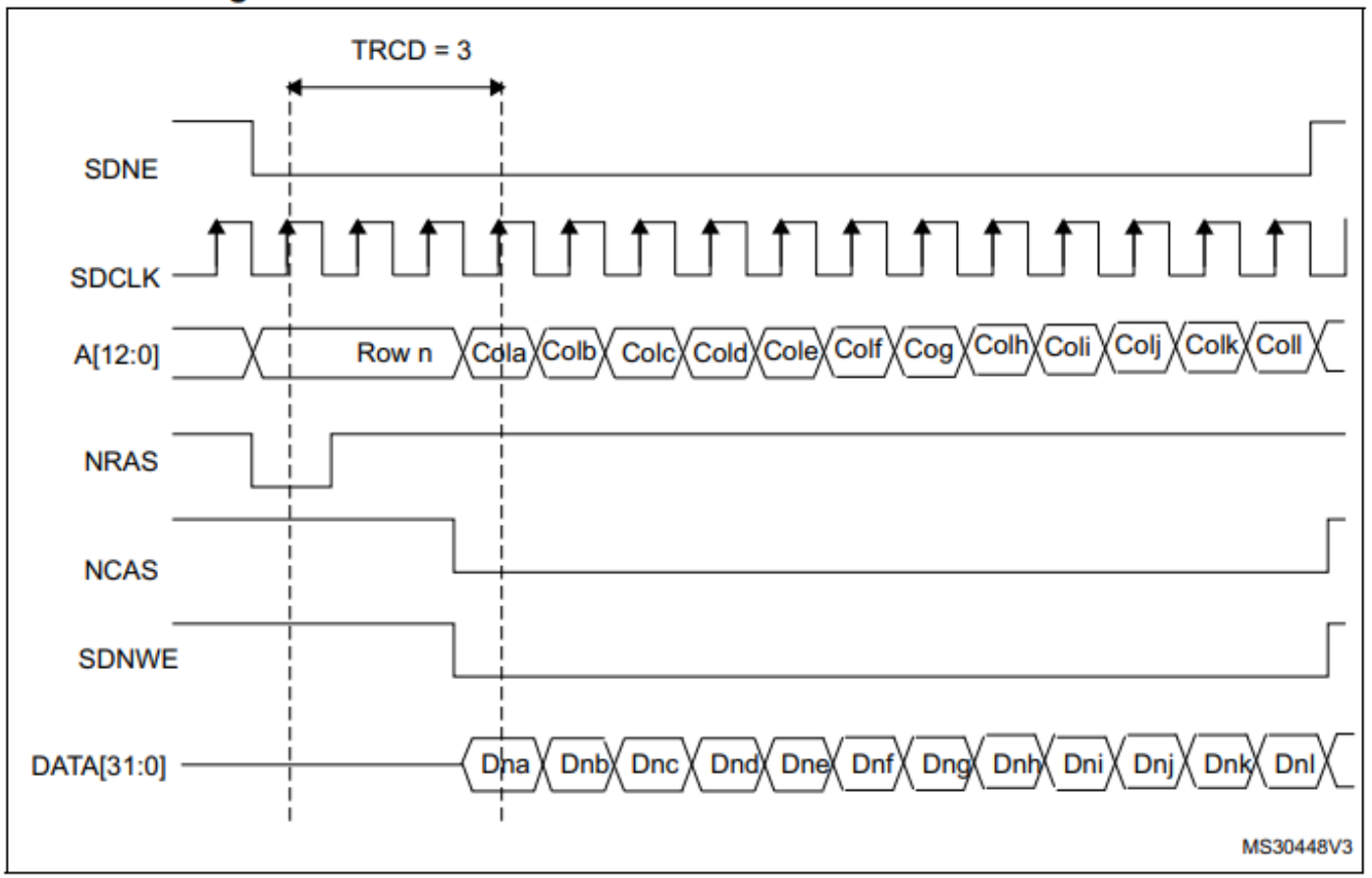


SDRAM Signals

SDRAM Seine

- SDRAM is synchronous: There is a clock signal
- RAS and CAS signals used to select between row and columns addresses
- Burst access modes keeps same row and changes column address

Figure 55. Burst write SDRAM access waveforms

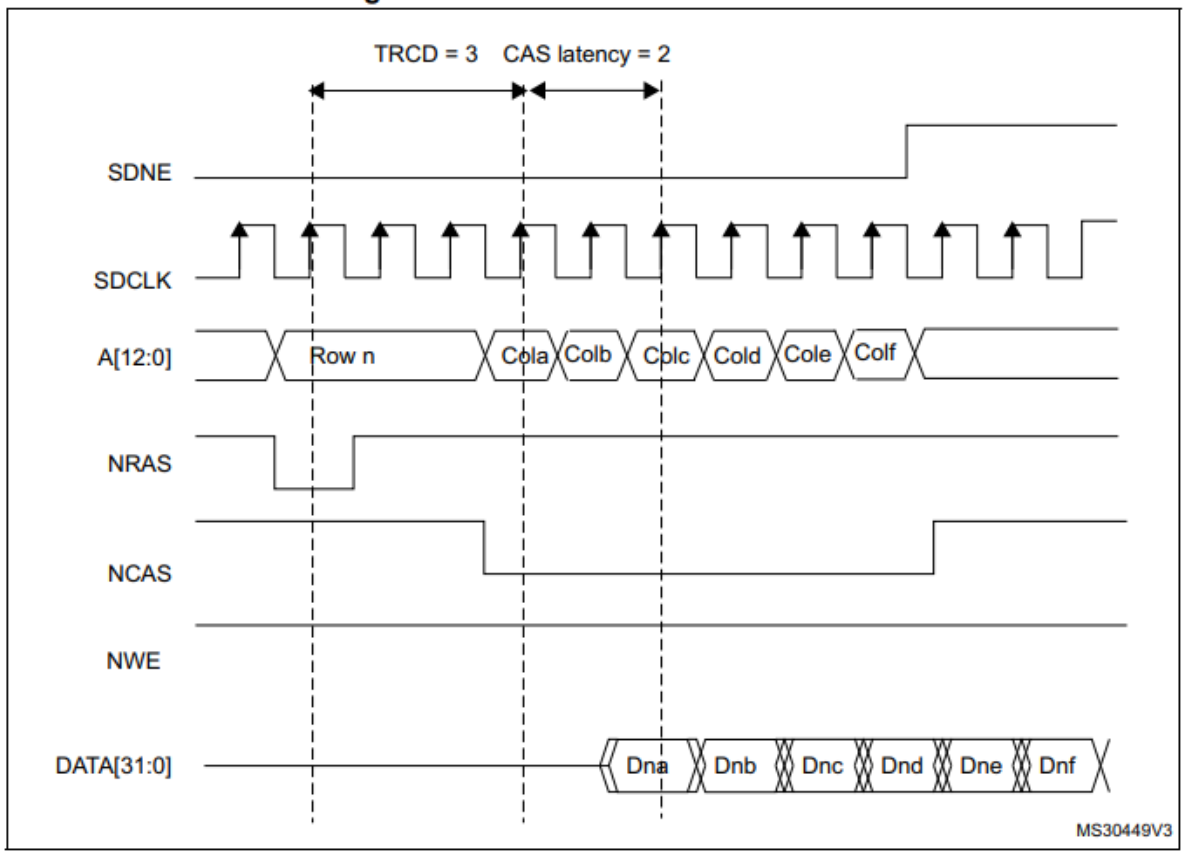


SDRAM Signals

SDRAM Seine

- SDRAM is synchronous: There is a clock signal
- RAS and CAS signals used to select between row and columns addresses
- Burst access modes keeps same row and changes column address

Figure 56. Burst read SDRAM access

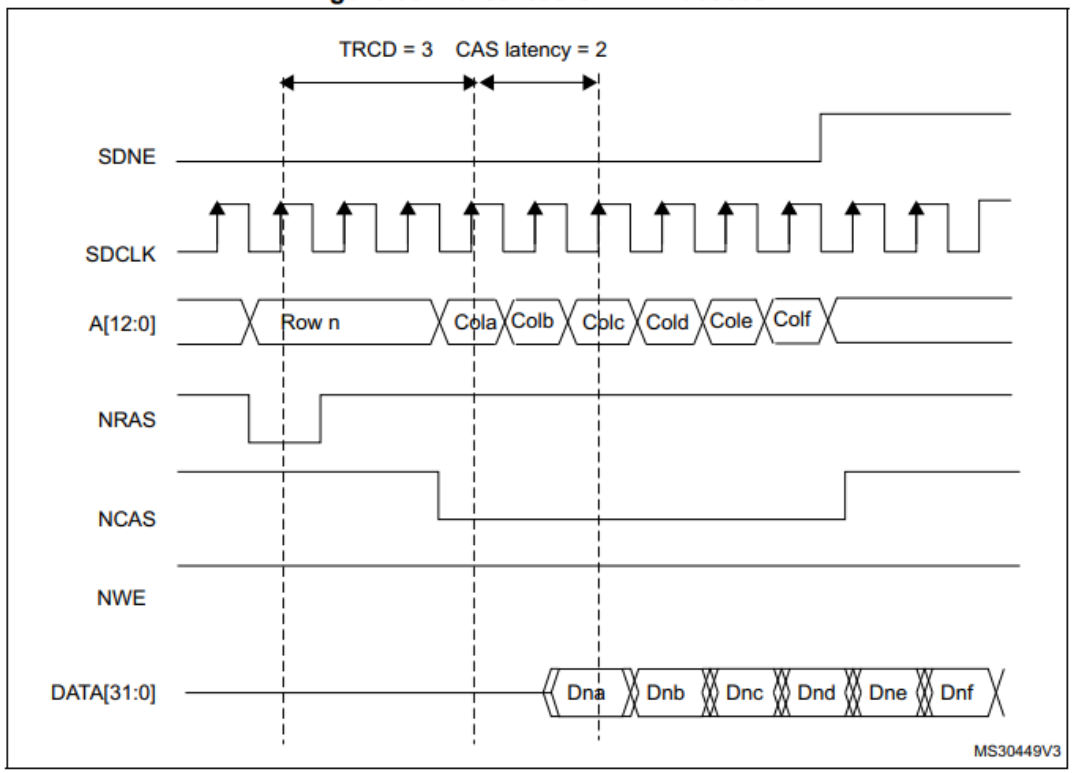


SDRAM Signals

SDRAM Seine

- Timing is crucial. Lots of timing parameters to set-up (in the external memory controller's registers)
  - CAS latency: The time between supplying a column address and then receiving back the corresponding data. Defined in terms of number of clock cycles
  - Read cycle time: The time between successive read operations to an open row.
  - Read the datasheet for the memory device

Figure 56. Burst read SDRAM access



Programming microcontroller for external memory access
Mikrobeheerder program vir eksterne geheue toegang

- Programming the microcontroller to access external memories involve:
- Setup registers in the external memory controller peripheral (FMC in our case). Specify:
  - Memory type
  - Number of address/data signals
  - Memory device timings
  - Other device specific commands and settings
  - Use HAL and device configuration tool
- Access external memory through normal memory read/write
  - LDR
  - STR

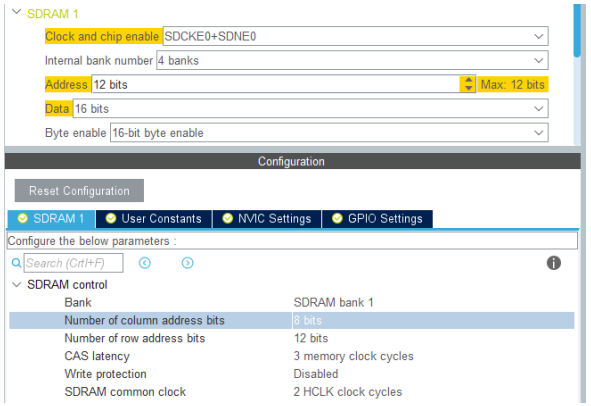


Figure 34. FMC memory banks