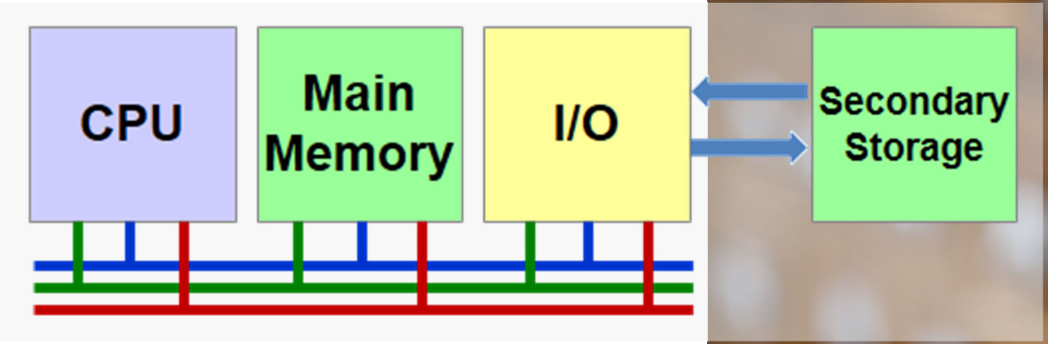
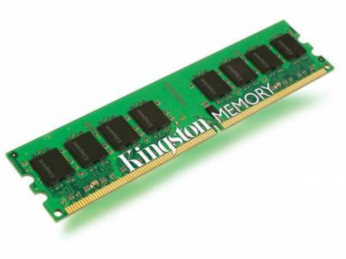
Memory

Computer Engineering 2

Motivation

Storing and retrieving data





Agenda

- **Memory Technologies**
  - PROM, EEPROM and flash, SRAM, SDRAM
- **On-CHIP Memories STM32F429ZISRAM**
  - SRAM and Flash
- **External Memory (Off-Chip)**
  - Flexible Memory Controller
- **Appendix: Trends and Figures**

Learning Objectives

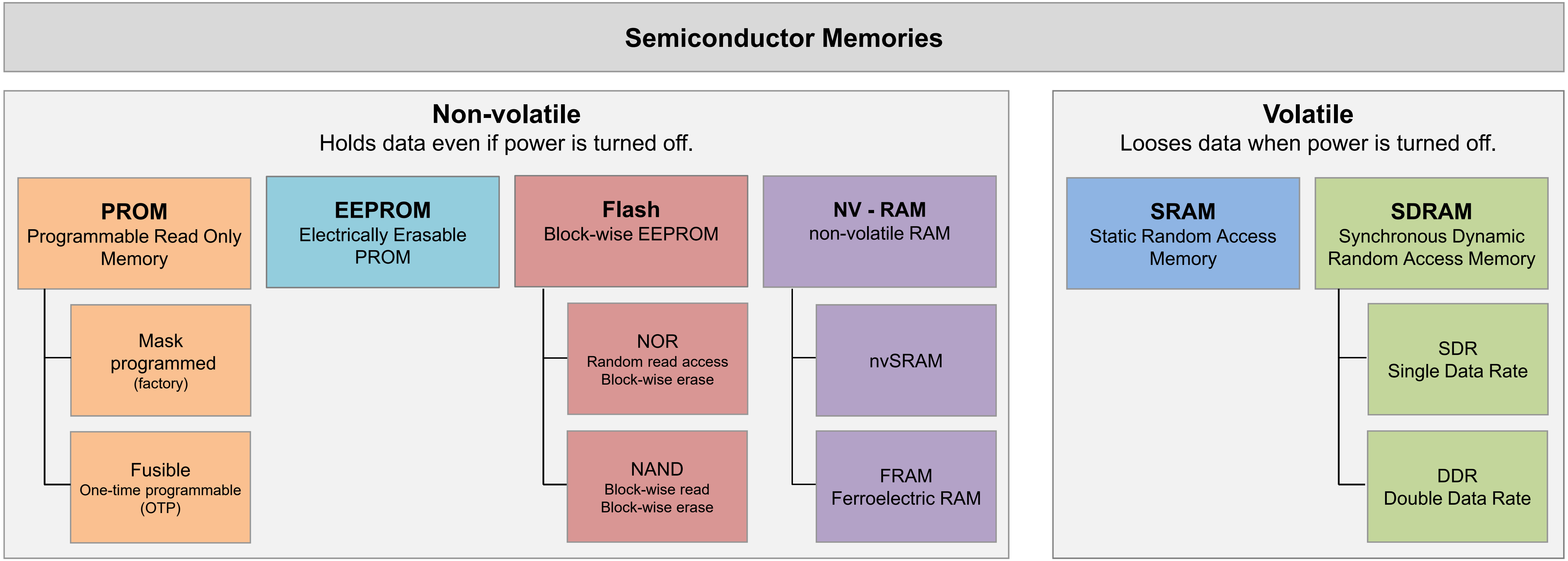
At the end of this lesson you will be able

- to classify widely used memory technologies
- to discuss the structure and function of an SRAM (static RAM)
- to discuss the structure and function of flash memory
- to outline the structure and function of an asynchronous SRAM device
- to outline how an external asynchronous SRAM device can be connected through the flexible memory controller (FMC)
- to explain how an internal 32-bit access is partitioned into several external half-word or byte accesses
- to interpret timing diagrams for read and write accesses to external, asynchronous SRAMs
- to summarize the differences between a NOR and a NAND flash
- to summarize the differences between a static RAM (SRAM) and a dynamic RAM (SDRAM)

Semiconductor Fundamentals

MEMORY TECHNOLOGIES

Memory Technologies



Semiconductor Memories

Non-volatile
Holds data even if power is turned off.

- **PROM**
  Programmable Read Only Memory
  - Mask programmed (factory)
  - Fusible
    One-time programmable (OTP)
- **EEPROM**
  Electrically Erasable PROM
- **Flash**
  Block-wise EEPROM
  - NOR
    Random read access
    Block-wise erase
  - NAND
    Block-wise read
    Block-wise erase
- **NV - RAM**
  non-volatile RAM
  - nvSRAM
  - FRAM
    Ferroelectric RAM

Volatile
Looses data when power is turned off.

- **SRAM**
  Static Random Access Memory
- **SDRAM**
  Synchronous Dynamic Random Access Memory
  - SDR
    Single Data Rate
  - DDR
    Double Data Rate

Units

Unit Symbols

- b = bit B = Byte

Memory Chips

- Binary prefixes according to JEDEC[1] and IEC[2]

Kilo	K	= 1024	
Mega	M	= 1024 x 1024	= 1'048'510
Giga	G	= 1024 x 1024 x 1024	= 1'073'741'824

Hard Disks

- Often use SI (or metric) prefixes

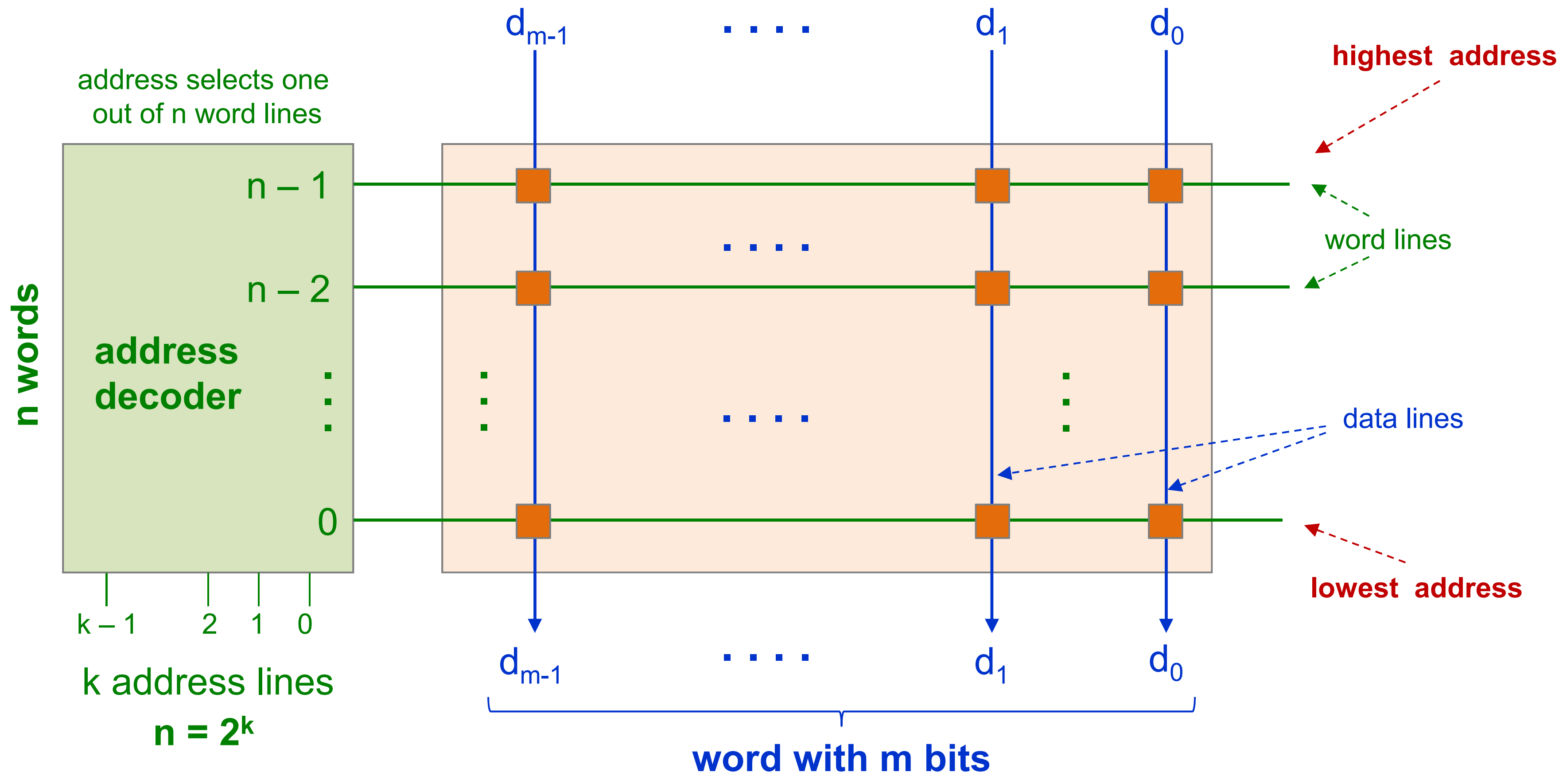
Kilo	k	= 1000
Mega	M	= 1000 x 1000
Giga	G	= 1000 x 1000 x 1000

[1] JEDEC Solid State Technology Association
[2] International Electrotechnical Commission

Memories Are Arrays of Bit Cells

- **Memory Architecture → n x m array**
  - n words with m data bits

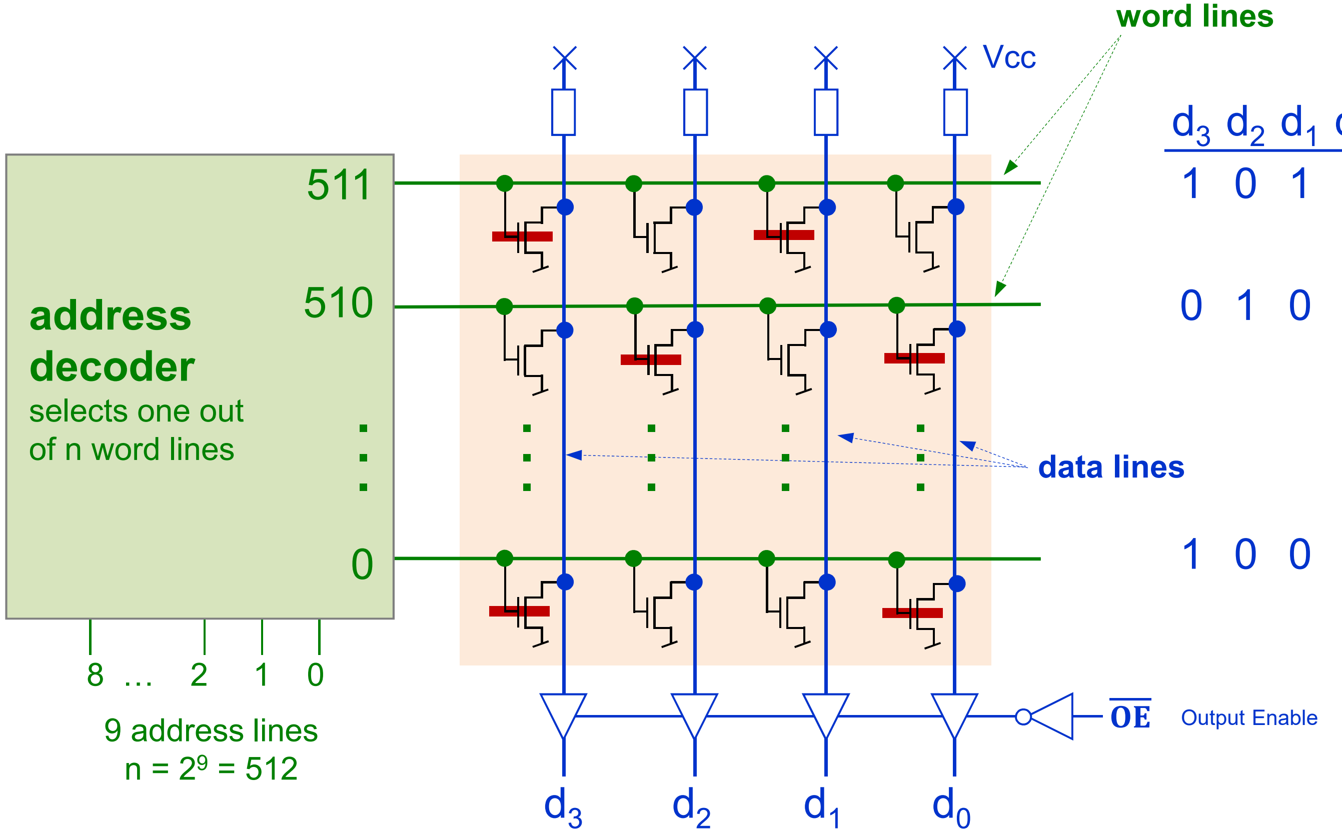
🟧 Bit cell → stores ‘1’ or ‘0’



PROM – Programmable Read Only Memory

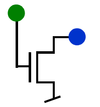
- **n x m array** → n addresses with m data bits Example 512 x 4 bit

n = number of word lines
m = number of bit lines



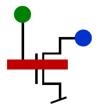
d_3	d_2	d_1	d_0
1	0	1	0
0	1	0	1
1	0	0	1

Fusible Transistors
Programming applies higher voltage to destroy transistors (blow fuses)
Process is **not** reversible



Word line = ‘1’
→ transistor shortens bit line to GND

Word line = ‘0’
→ transistor is open; pull-up pulls bit line to Vcc

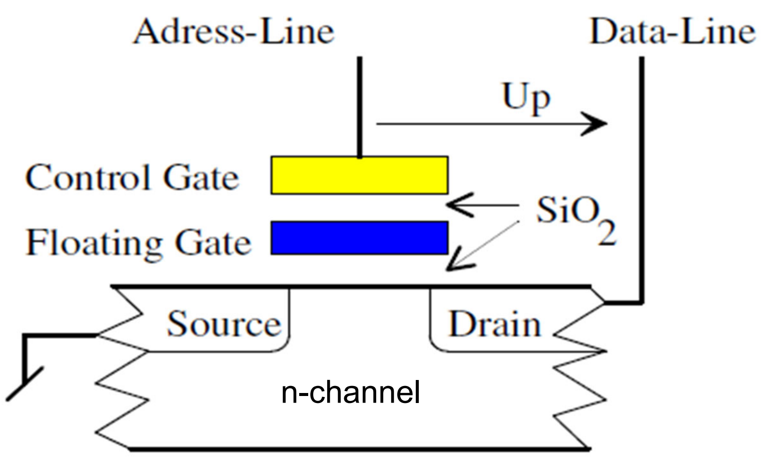


Transistor destroyed (fused)
→ always open, i.e. pull-up pulls bit line to Vcc

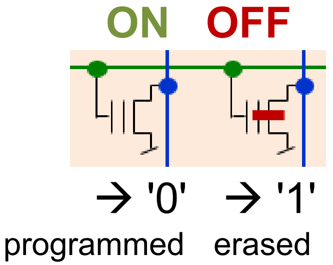
EEPROM and Flash

Making PROMs Reprogrammable

- "Floating Gate" transistor
  - Replace fusing by reprogrammable “Floating Gate”
- Write cell to '0‘ → **ON**
  - High voltage Up deposits charge on floating gate (isolated by SiO2)
  - Transistor ON (conducting) if control gate equal '1‘
- Erase cell to '1‘ → **OFF**
  - Discharge floating gate with negative Up
  - Transistor is OFF, i.e. blocking independent of value on control gate
- EEPROM
  - High cell area → low density, high cost per bit
- Flash
  - Erasing can only be done for whole sectors → small cell area, high density, low cost per bit

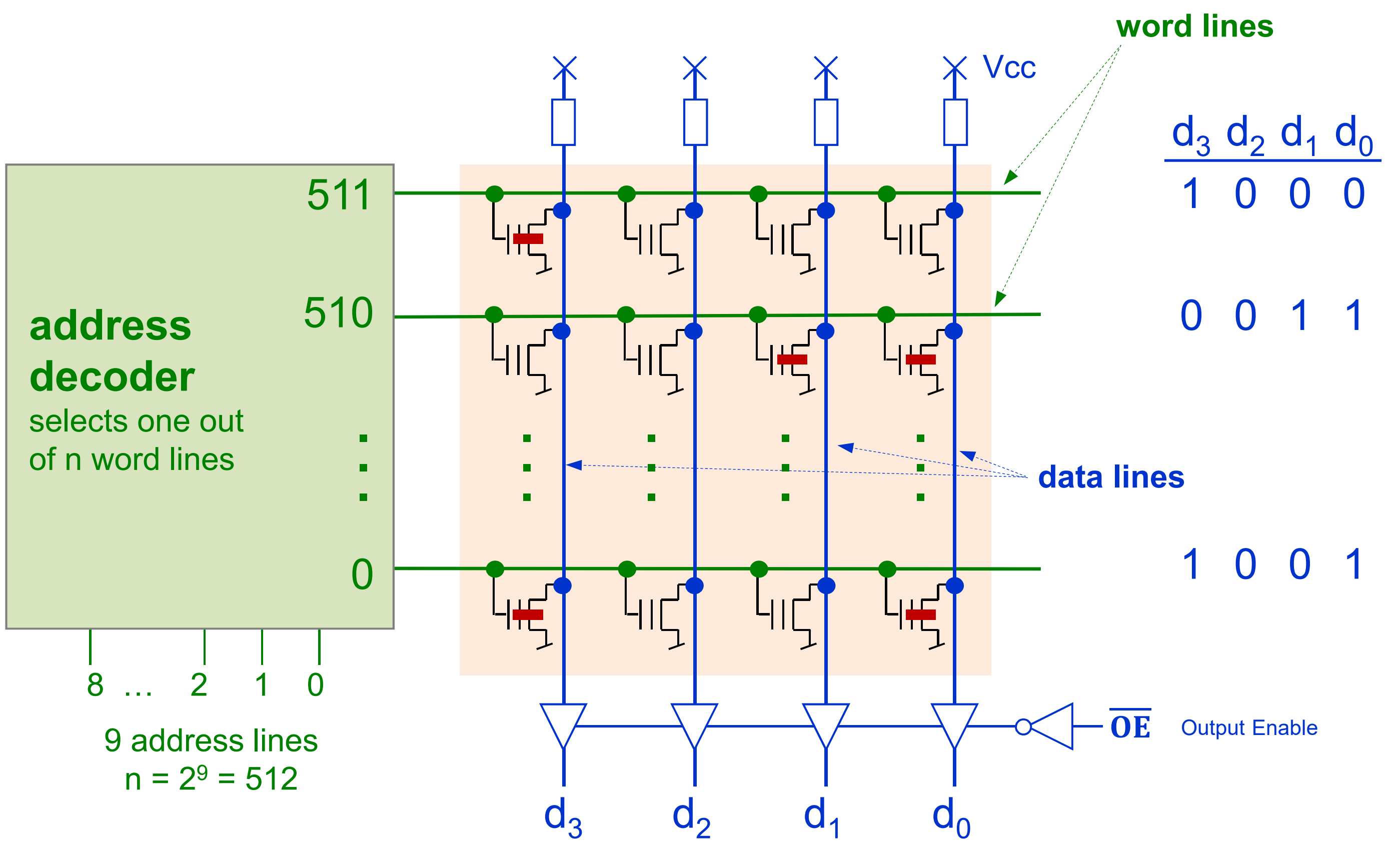


Silicon dioxide (SiO2) is an excellent isolator

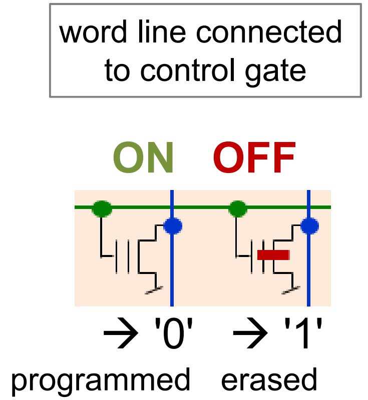


EEPROM and Flash

- **Use ‘Floating Gates’ instead of ‘Fusible Transistors’**



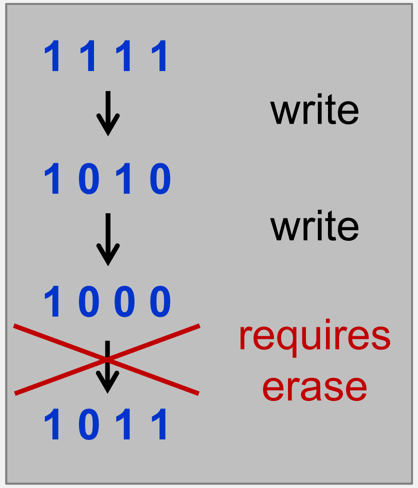
d_3	d_2	d_1	d_0
1	0	0	0
0	0	1	1
1	0	0	1



Flash

Write Operations (Programming)

- Can only change bits from '1' to '0'
  - Otherwise an erase operation is required
- Word, half-word or byte access possible
- Writing a double word ~16 us
  - I.e. around 1000 times slower than SRAM



1 1 1 1

write

1 0 1 0

write

1 0 0 0

requires erase

1 0 1 1

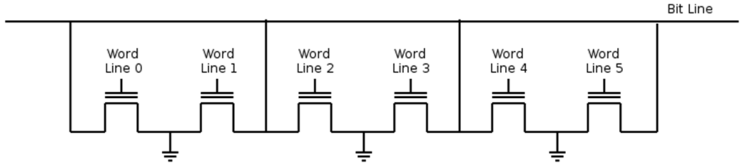
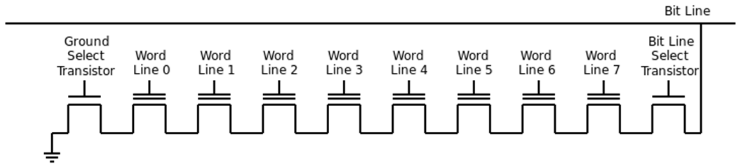
Erase Operations

- Change all bits from '0' to '1'
  - Only possible by sector or by bank, not on a word
  - Typical sector sizes of 16
- Erase of a 128 Kbytes sector takes between 1 and 2 seconds [1)]
- Endurance: 10'000 erase cycles [2)]
- Sector may not be accessed (write or read) during erase
  - I.e. execute program from another sector or from SRAM during erase

1) Depending on supply voltage and configuration parameters

2) Value from STM32F429ZI datasheet

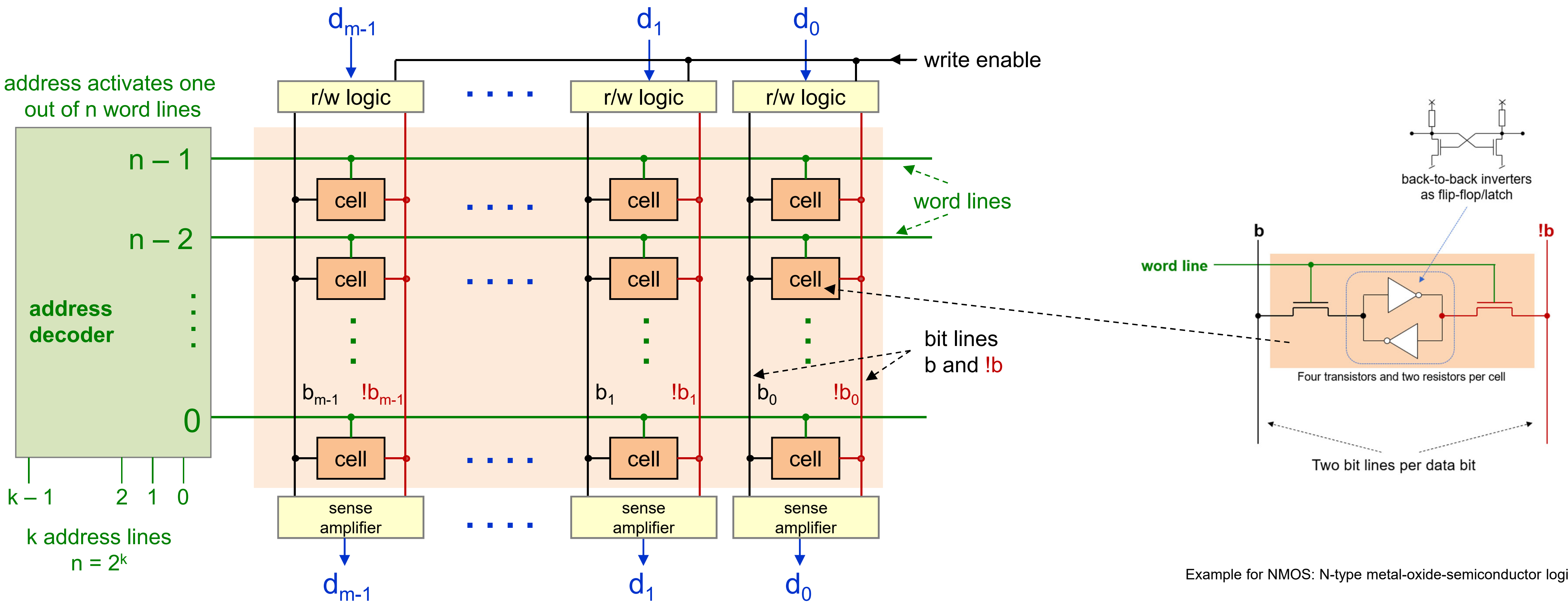
Flash – NOR vs NAND Topology

	NOR Flash	**NAND Flash**
Topology	 Bit Line Word Line 0 Word Line 1 Word Line 2 Word Line 3 Word Line 4 Word Line 5	 Bit Line Ground Select Transistor Word Line 0 Word Line 1 Word Line 2 Word Line 3 Word Line 4 Word Line 5 Word Line 6 Word Line 7 Bit Line Select Transistor
Applications	- **Execute code directly from memory** - **Persistent device configurations (replacement of EEPROM)**	- **File-based IO, disks** - **Large amounts of sequential data (images, SD cards, SSD)** - **Load programs into RAM before executing**
Density	- Medium Up to 2 GBit = 256 MByte	- High Up to 1 Tbit
Interface	- Read same as asynchronous SRAM - Types with serial interface available	- Special NAND flash interface - Error correction for defective blocks
Access	- Random access read ~0.12 µs - Writing individual bytes possible - Slow writes ~180 µs / 32 Byte	- Slow random access read: 1. Byte 25 µs, then 0.03 µs each - Writing of individual bytes difficult - Fast block write ~300 µs / 2'112 Bytes

source of drawings: Wikipedia

SRAM – Static Random Access Memory

- **n x m SRAM Architecture → flip-flop (latch) based cells**



Example for NMOS: N-type metal-oxide-semiconductor logic

SRAM – Static Random Access Memory

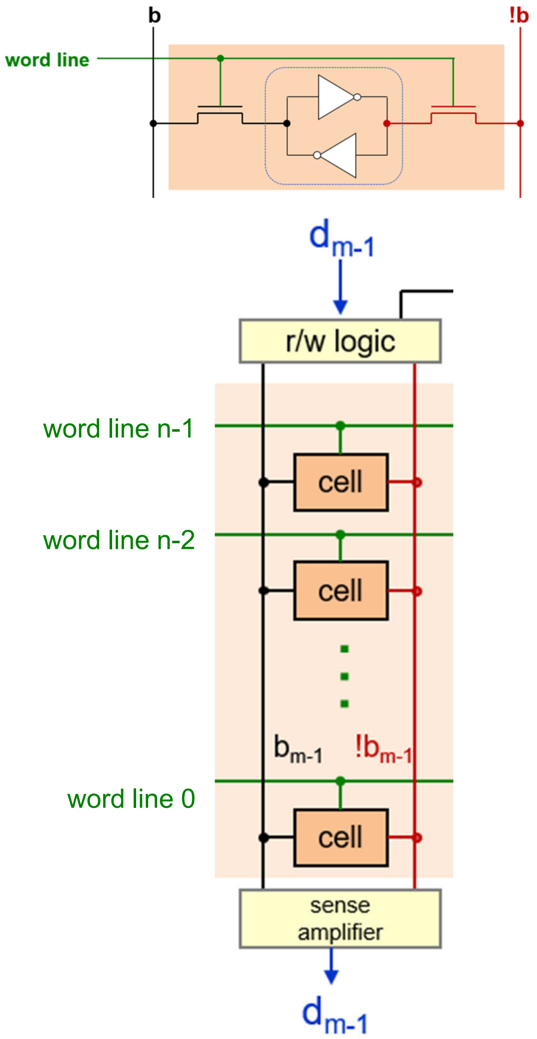
Writing a Row (Word)

Set bit lines b and !b to (1, 0) or (0, 1) respectively

Set the addressed word line to 1

Data is stored in cells

Set word line to 0



Reading a Row (Word)

Pre-charge both bit lines b and !b to 1

Briefly set word line to 1

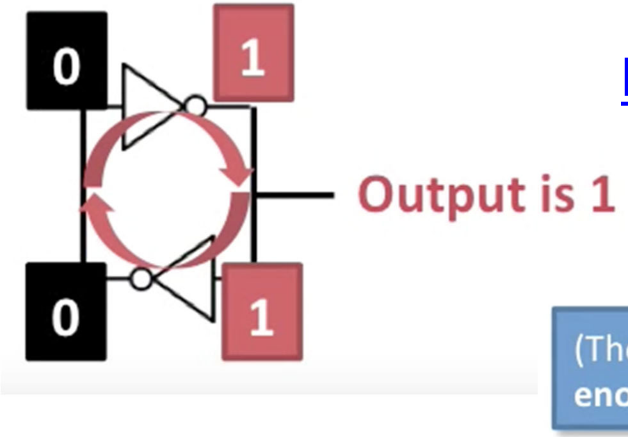
Inverters pull either b or !b towards (not to) ground

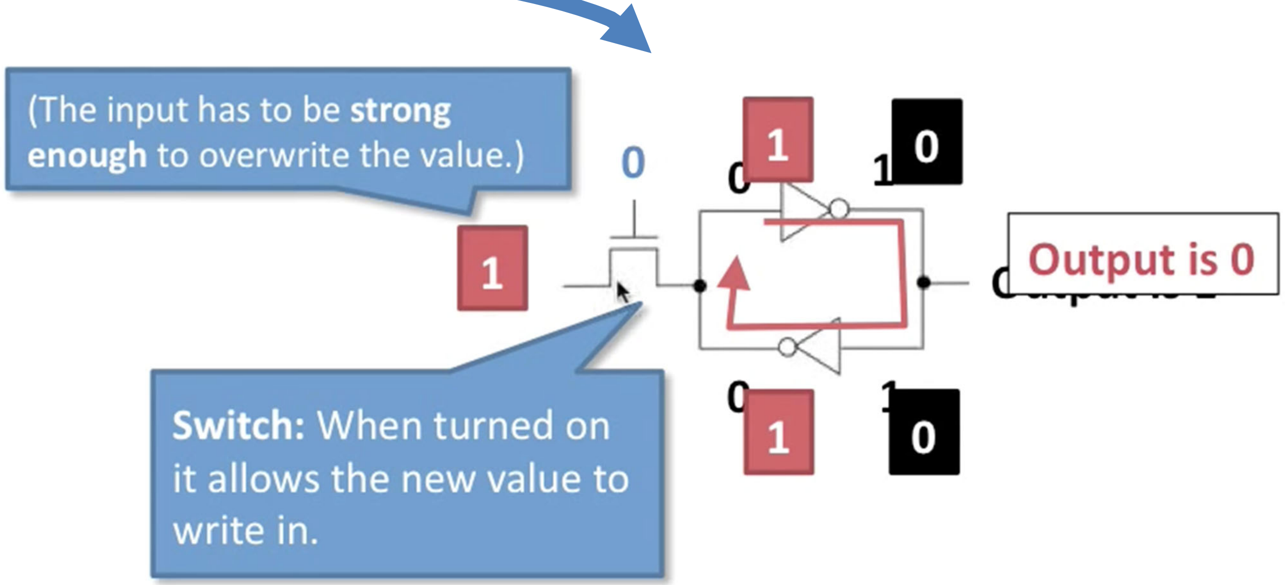
Sense amplifier amplifies small voltage difference between lines b and !b

SRAM – Static Random Access Memory

- **Structure of SRAM cell in NMOS[1]**
  - Flip-flop (latch) based structure, change from ‘1’ to ‘0'

https://www.youtube.com/watch?v=mwNqzc1o5zM





(The input has to be **strong enough** to overwrite the value.)

Switch: When turned on it allows the new value to write in.

[1] N-type metal-oxide-semiconductor logic

SRAM – Static Random Access Memory

- **Read and write**
  - All accesses take roughly the same time
  - Access time independent of location of data item in memory
  - Access time independent of previous access[1)]

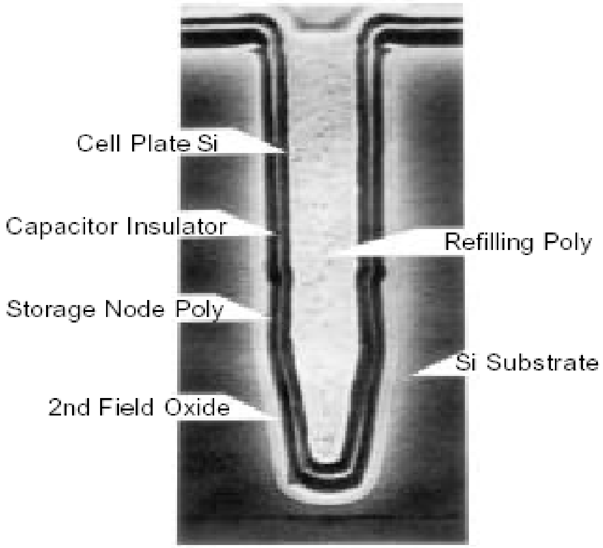
- **Volatile**
  - Memory content retained only as long as device is powered

- **Static**
  - Storage elements similar to flip-flops / latches
  - <u>No</u> refresh required
    - refresh: periodic reading and rewriting of memory cell to maintain the content

[1)] as opposed to a DRAM that favors burst accesses

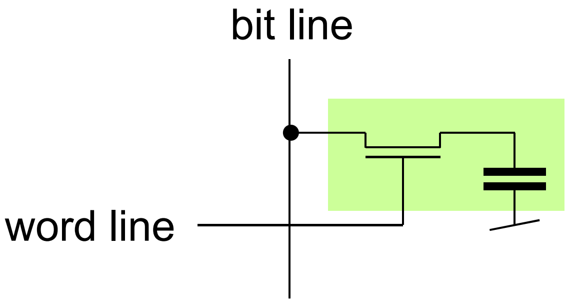
SDRAM – Synchronous Dynamic RAM

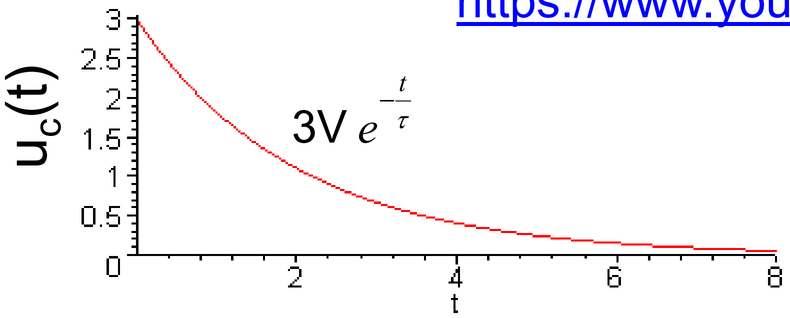
- **Synchronous Dynamic Random Access Memory**
  - Information stored as charge in capacitor
  - High integration
    - Large memories at low cost
    - Allows to store large amounts of data
  - Leakage current → Loss of charge
    - Capacitor holds charge only for a few milliseconds
    - Charge has to be refreshed periodically → dynamic
    - Refresh logic usually located on SDRAM device



Trench Cell

https://www.youtube.com/watch?v=3s7zsLU83bY



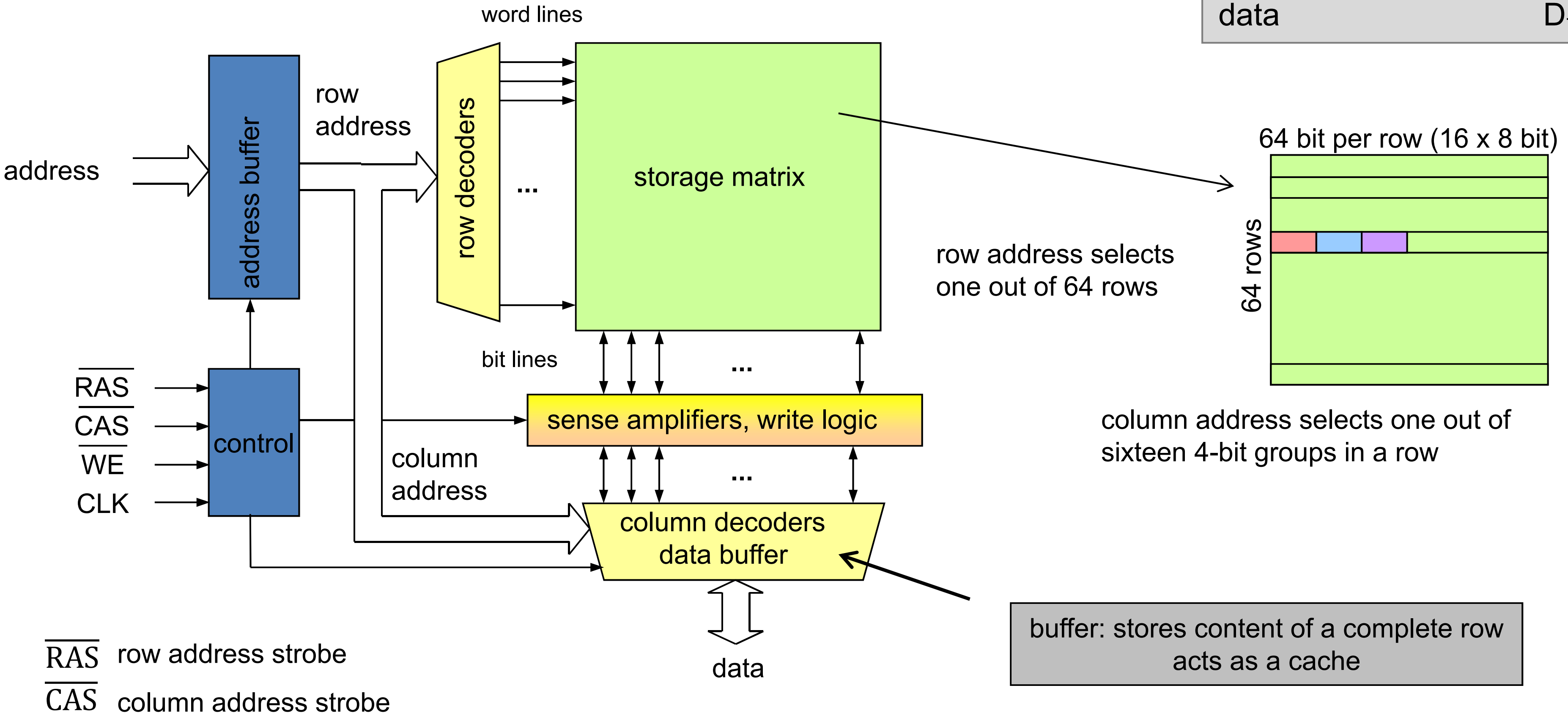


SDRAM – Synchronous Dynamic RAM

SDRAM Structure

- Row and column addresses multiplexed

Organization e.g.	512 x 8-bit
storage matrix	64 x 64 bit
row address	$A_5 .. A_0$
column address	$A_2 .. A_0$
data	$D_7 .. D_0$



row address selects one out of 64 rows

column address selects one out of sixteen 4-bit groups in a row

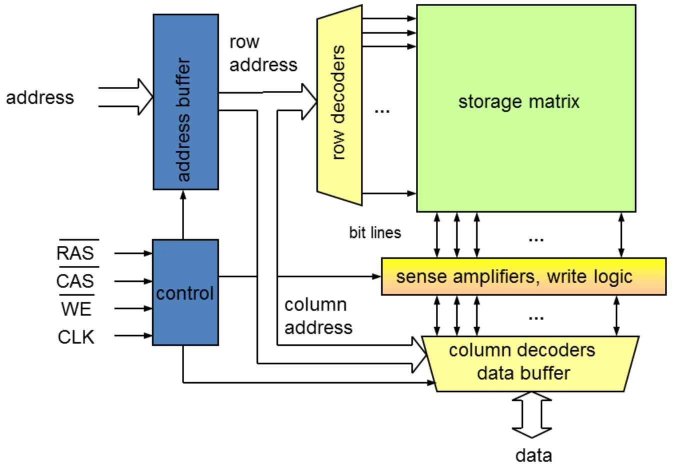
buffer: stores content of a complete row acts as a cache

$\overline{RAS}$ row address strobe

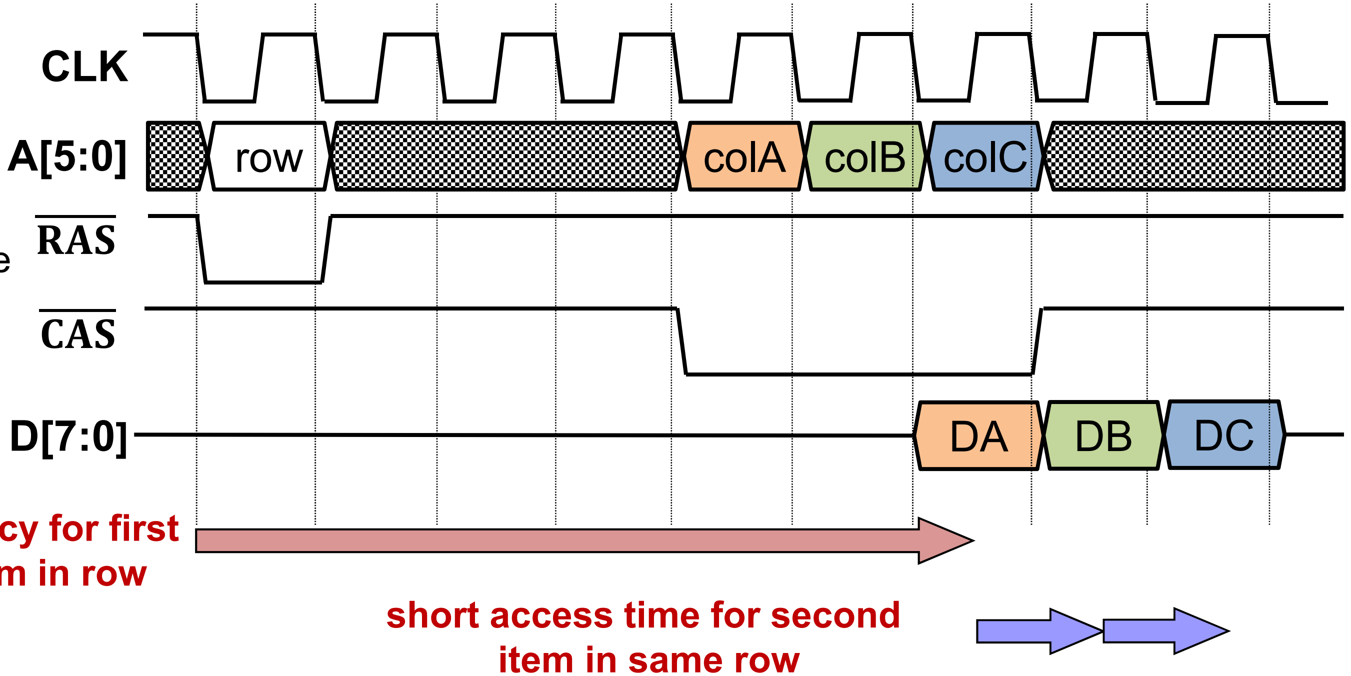
$\overline{CAS}$ column address strobe

SDRAM – Synchronous Dynamic RAM

- **Synchronous Interface**
  - Multiplexed row and column addresses
  - Clocks up to 1200 MHz



read cycle on single data rate SDRAM



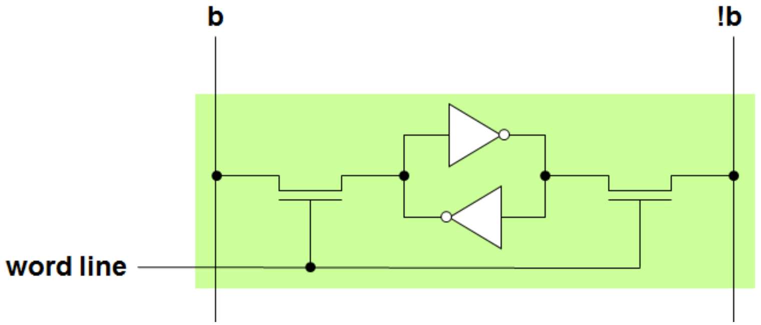
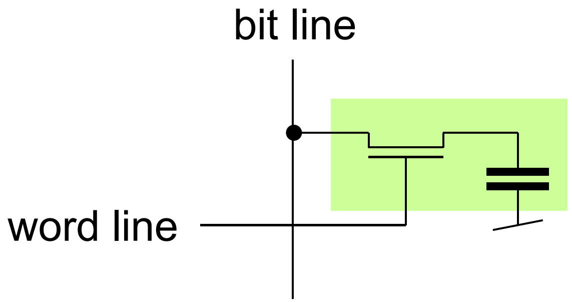
long latency for first data item in row

short access time for second item in same row

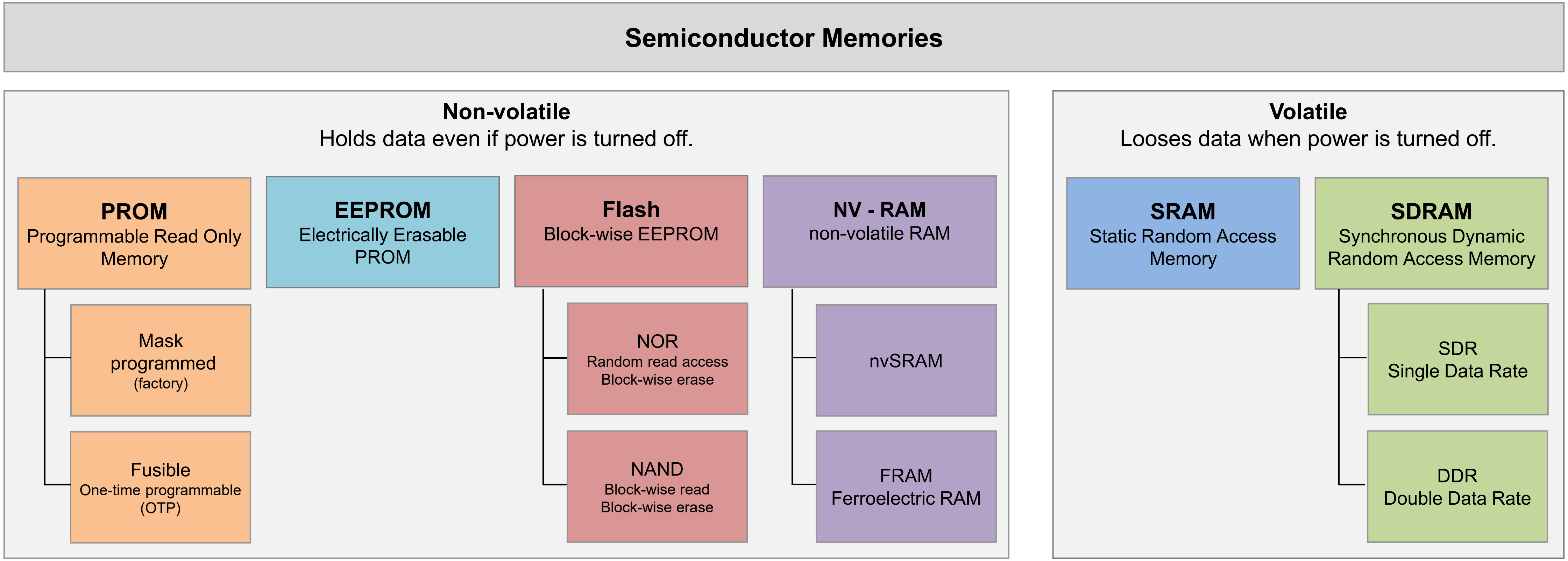
$\overline{RAS}$ low → The master places the 6-bit row address on lines A[5:0].

$\overline{CAS}$ low → The master places the 3-bit column address on lines A[2:0]. Lines A[5:3] are unused.

SDRAM – Synchronous Dynamic RAM

Static RAM (SRAM)	Synchronous Dynamic RAM (SDRAM)
Flip-flop/latch → 4 Transistors / 2 resistors  	Transistor and capacitor 
Large cell • Low density, high cost • Up to 64 Mb per device	Small cell • High density, low cost • Up to 4 Gb per device
Almost no static power consumption • Static i.e. no accesses taking place	Leakage currents • Requires periodic refresh
Asynchronous interface (no clock) • Simple connection to bus	Synchronous interface (clocked) • Requires dedicated SDRAM Controller
All accesses take roughly the same time • ~5ns per access → 200 MHz • Suitable for distributed accesses	Long latency for first access of a block • Fast access for blocks of data (bursts) • Large overhead for single byte

Memory Technologies



Semiconductor Memories

Non-volatile
Holds data even if power is turned off.

PROM
Programmable Read Only Memory

- Mask programmed (factory)
- Fusible
  One-time programmable (OTP)

EEPROM
Electrically Erasable PROM

Flash
Block-wise EEPROM

- NOR
  Random read access
  Block-wise erase
- NAND
  Block-wise read
  Block-wise erase

NV - RAM
non-volatile RAM

- nvSRAM
- FRAM
  Ferroelectric RAM

Volatile
Looses data when power is turned off.

SRAM
Static Random Access Memory

SDRAM
Synchronous Dynamic Random Access Memory

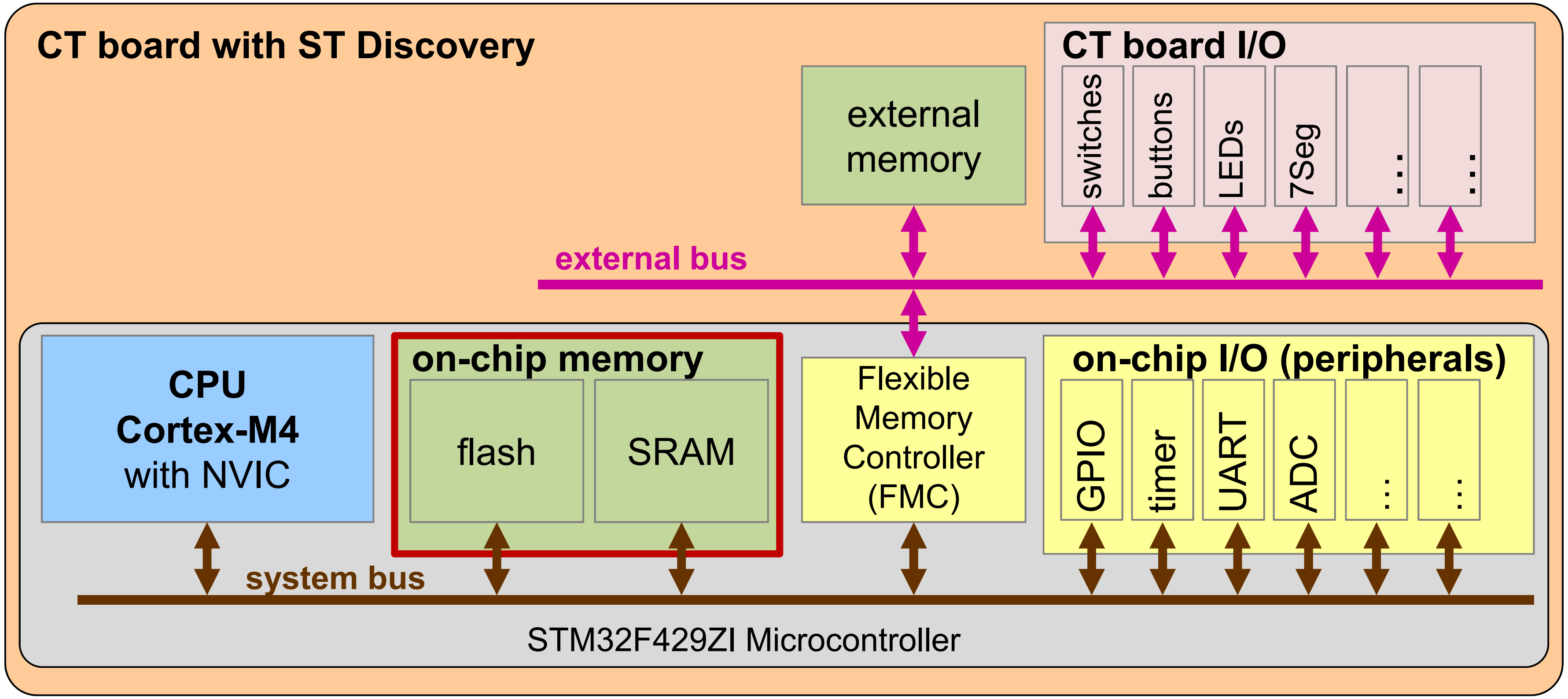
- SDR
  Single Data Rate
- DDR
  Double Data Rate

Our System

ON-CHIP MEMORIES STM32F429ZI

CT System Overview

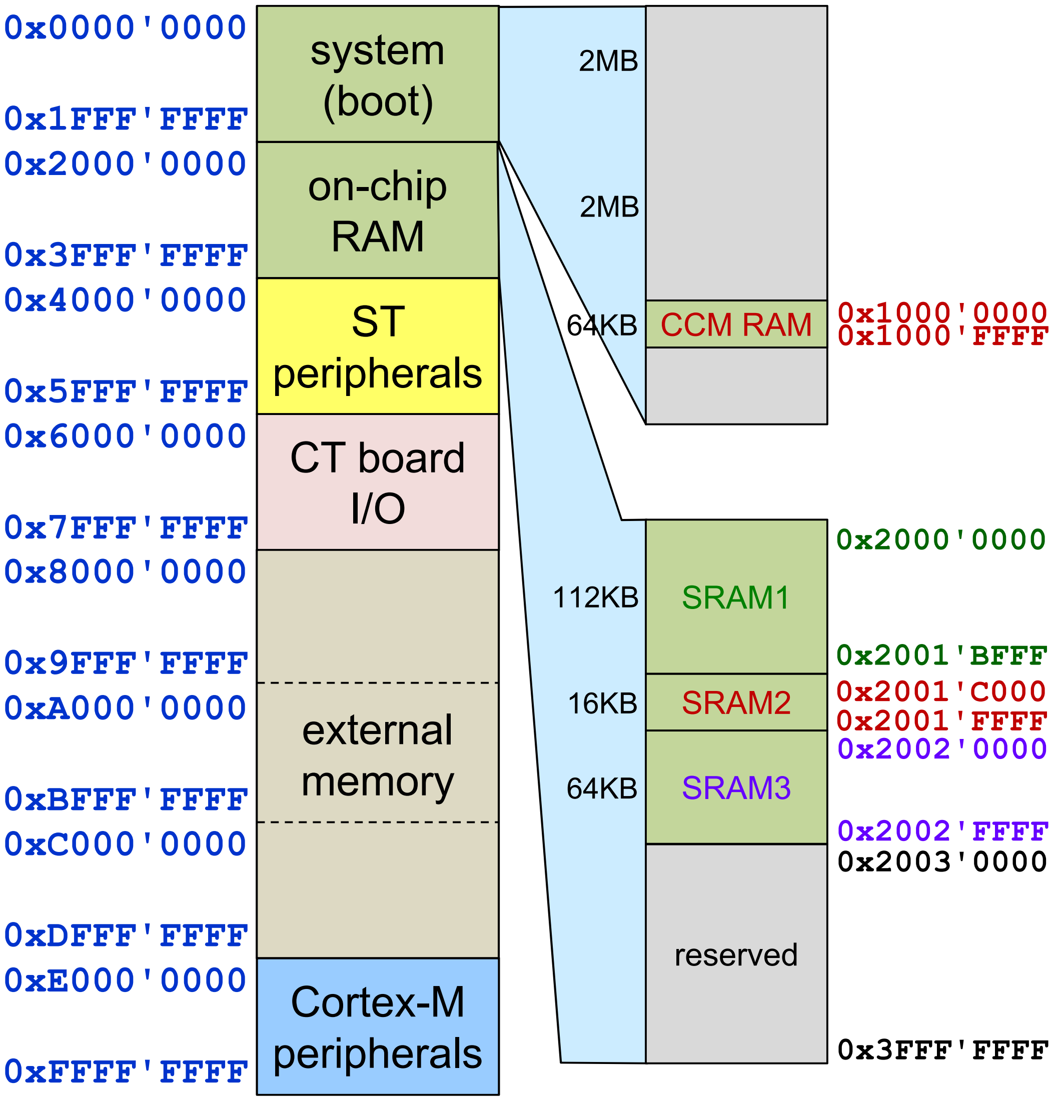
- **Simplified Model STM32F429ZI**
  - On-chip system bus 32 data lines, 32 address lines and control signals



On-chip Memory: SRAM

- **Address Regions**

SRAM1	112K bytes
SRAM2	16K bytes
SRAM3	64K bytes
CCM	64K bytes



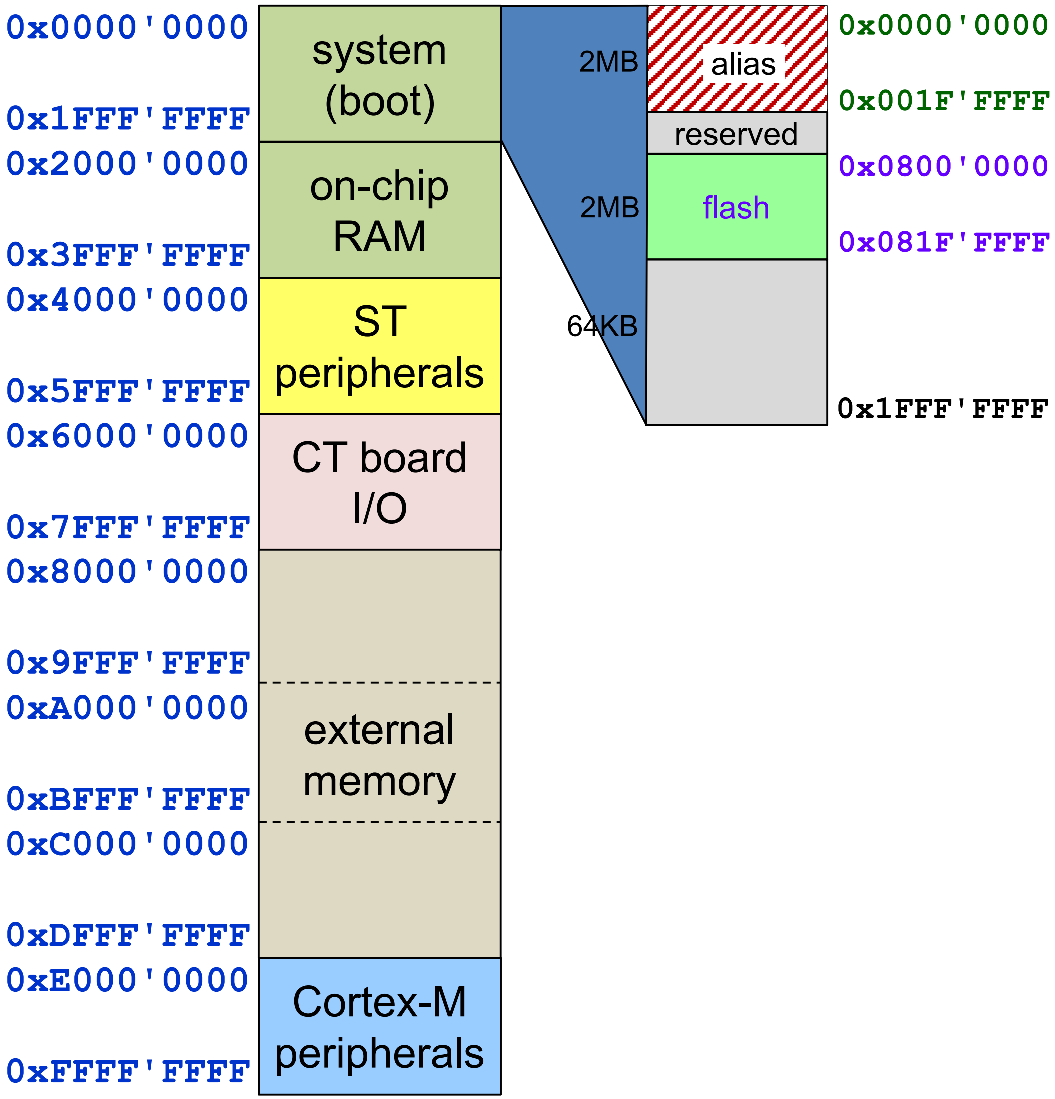
CCM: Core Coupled Memory –
Fast memory exclusively addressable by the CPU.

On-chip Memory: Flash

- **Flash**
  - Non-volatile memory
    - Memory content retained after power off
  - Store code and persistent data
  - NOR topology
    - Like most on-chip flash memories

Persistent Data denotes information that is infrequently accessed and not likely to be modified.

Source: Wikipedia



On-chip Memory: Flash

- **Flash Is Partitioned into Sectors**
  - Sectors can only be erased as a whole
  - Writing through control registers – no direct memory write accesses

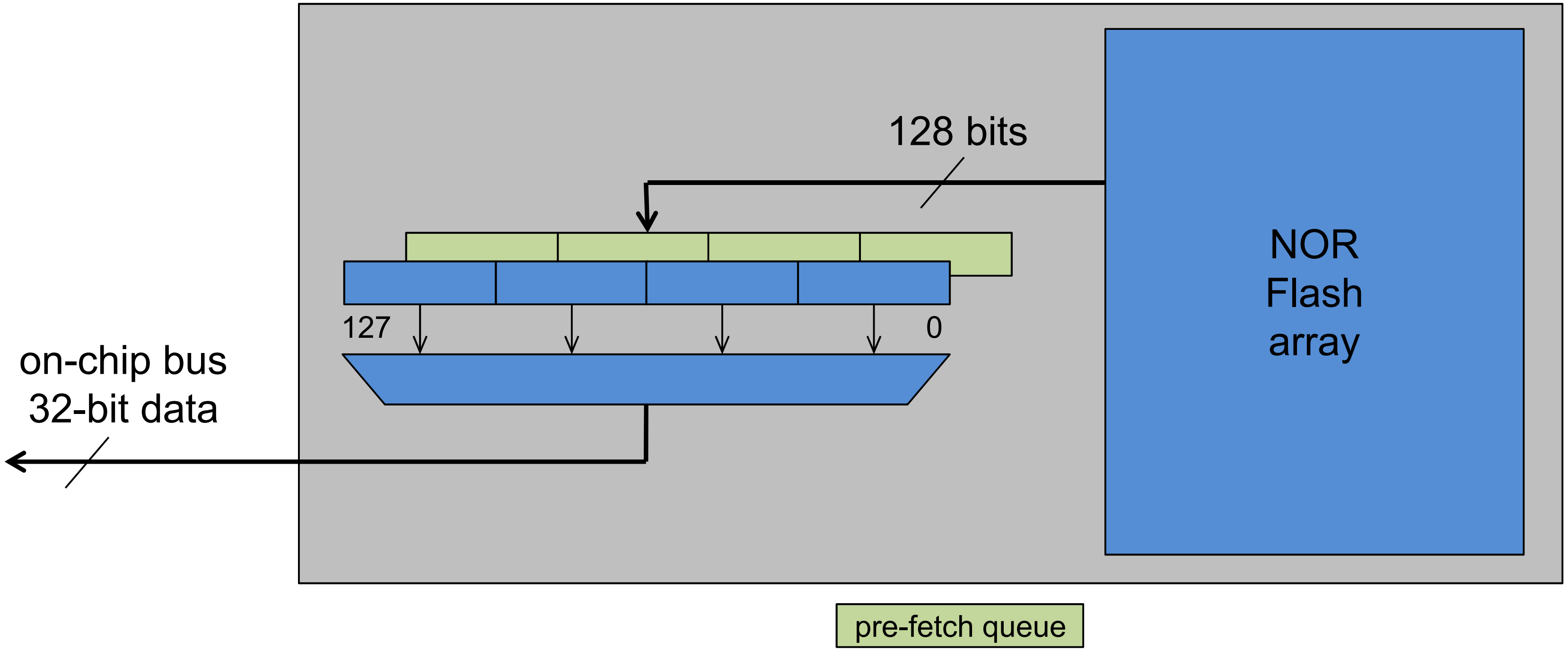
STM32F429ZI

Bank 1	Sector 0	0x0800'0000 – 0x0800'3FFF	16 Kbytes	total 2 Mbytes
	Sector 1	0x0800'4000 – 0x0800'7FFF	16 Kbytes	
	Sector 2	0x0800'8000 – 0x0800'BFFF	16 Kbytes	
	Sector 3	0x0800'C000 – 0x0800'FFFF	16 Kbytes	
	Sector 4	0x0801'0000 – 0x0801'FFFF	64 Kbytes	
	Sector 5	0x0802'0000 – 0x0803'FFFF	128 Kbytes	
	...	...	...	
	Sector 11	0x080E'0000 – 0x080F'FFFF	128 Kbytes	
Bank 2	Sector 12	0x0810'0000 – 0x0810'3FFF	16 Kbytes	
	Sector 13	0x0810'4000 – 0x0810'7FFF	16 Kbytes	
	Sector 14	0x0810'8000 – 0x0810'BFFF	16 Kbytes	
	Sector 15	0x0810'C000 – 0x0810'FFFF	16 Kbytes	
	Sector 16	0x0811'0000 – 0x0811'FFFF	64 Kbytes	
	Sector 17	0x0812'0000 – 0x0813'FFFF	128 Kbytes	
	...	...	...	
	Sector 23	0x081E'0000 – 0x081F'FFFF	128 Kbytes	

On-chip Memory: Flash

Flash Has Higher Latency

- Read requires up to 8 Wait States[1] on on-chip bus
- ST uses 128-bit buffer with pre-fetch queue
  - Reduces performance penalty when executing sequential instructions



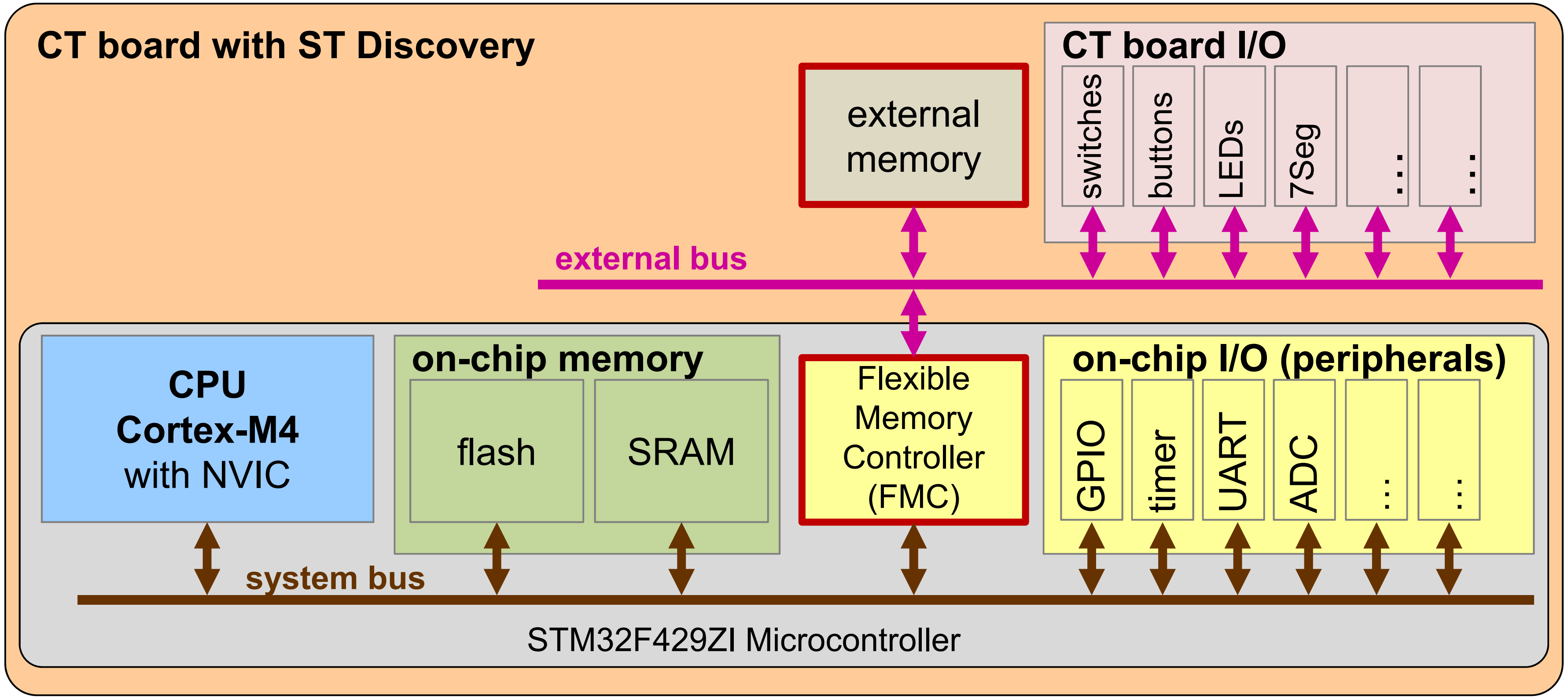
[1] depending on clock frequency and supply voltage

Extending Our System

EXTERNAL MEMORY (OFF-CHIP)

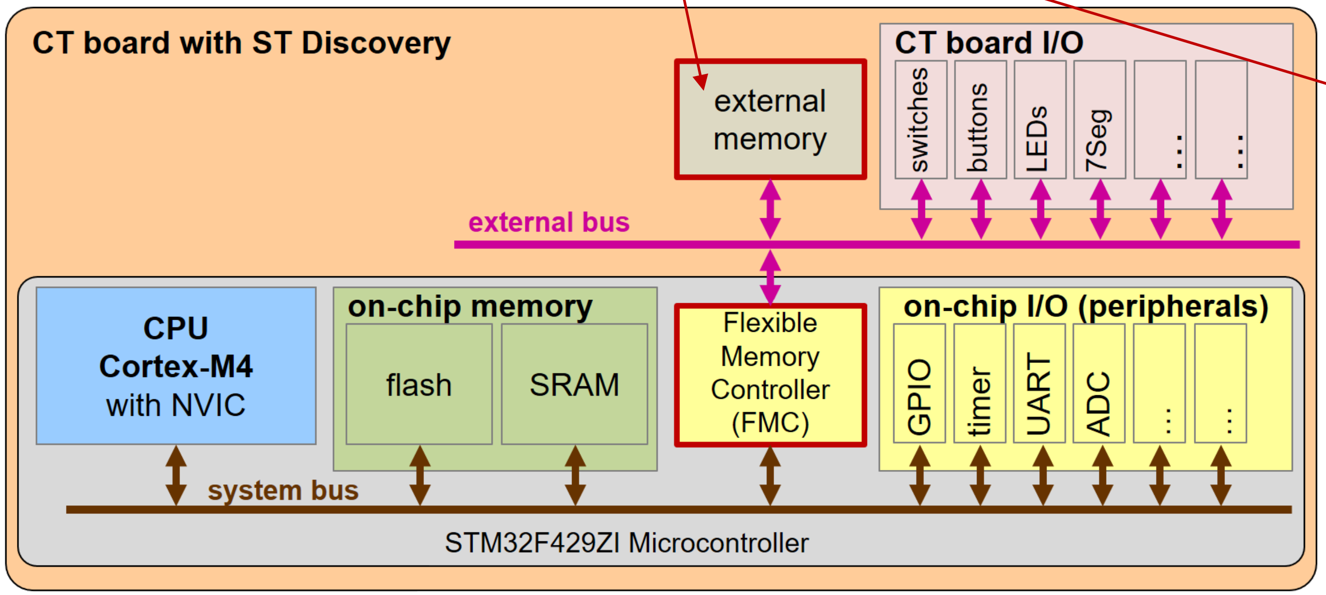
CT System Overview

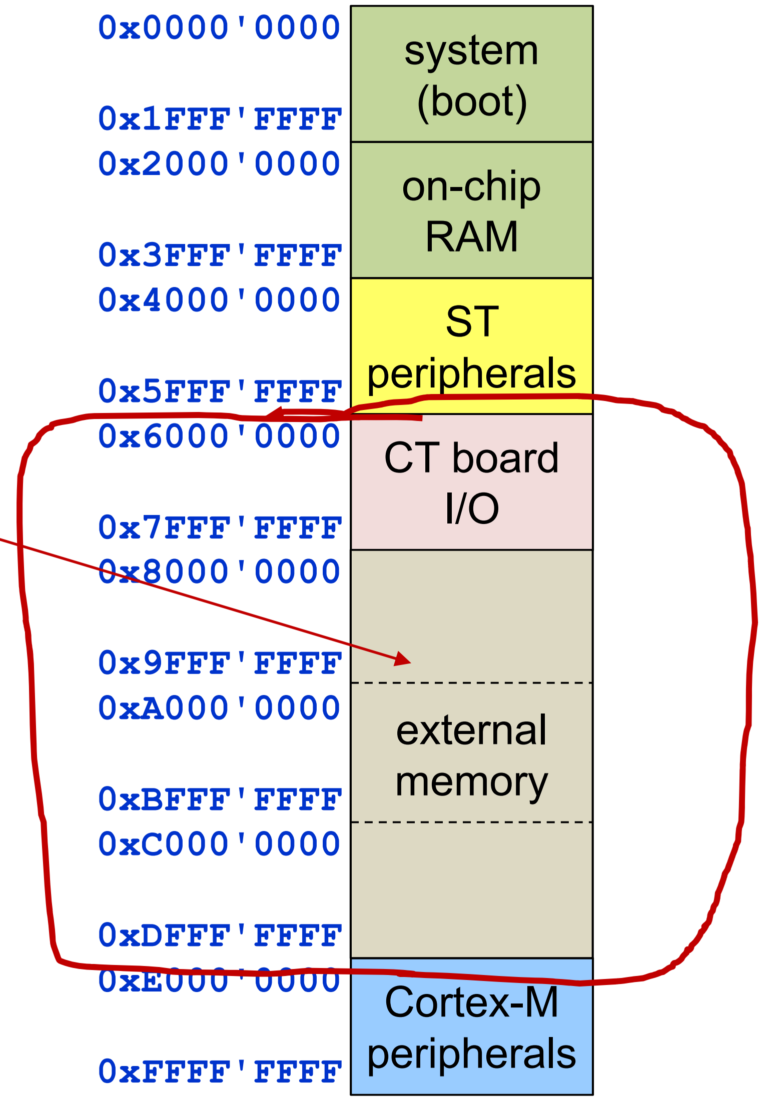
- **Simplified Model STM32F429ZI**
  - On-chip system bus 32 data lines, 32 address lines and control signals
  - External bus 16 data lines, 26 address lines and control signals



External Memory

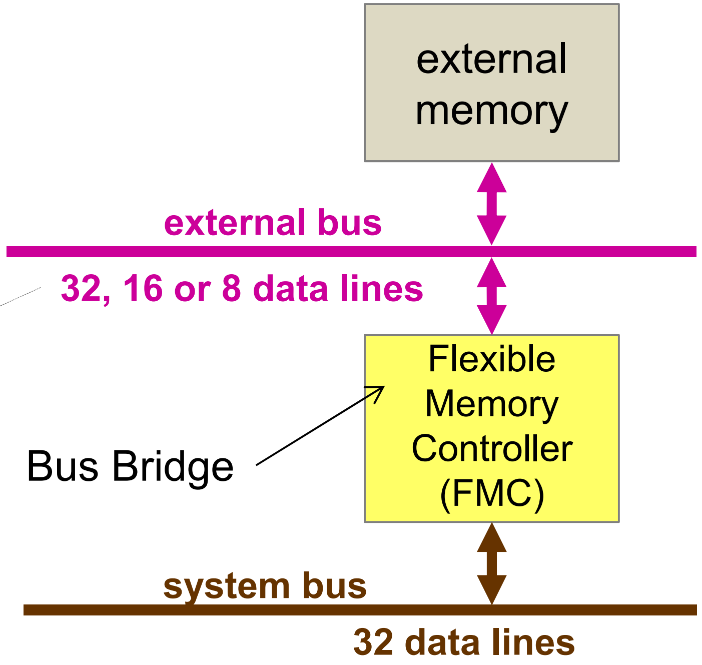
- **Extend On-chip Memory**
  - Flexible Memory Controller (FMC)
    - SRAM and NOR flash
    - Synchronous DRAM
    - NAND flash





Flexible Memory Controller

- **FMC – Configurable Bus Bridge**
  - Bridge between system bus and external bus
    - Slave on system bus
    - Master on external bus
  - System bus accesses
    - In address range 0x6000'0000 to 0xDFFF'FFFF
    - Bridged to external bus
    - I.e. FMC initiates a bus cycle on external bus



Number of data lines is a design decision and depends on the external memory device

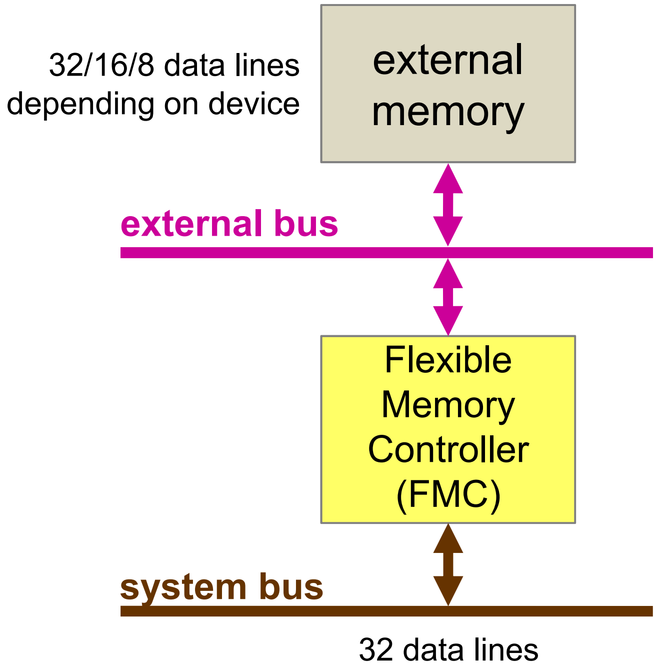
Flexible Memory Controller

Different Number of Data Lines Causes Bottleneck

- E.g. 32-bit (word) access to 8-bit external memory [1]
  - Single access on system bus
  - Results in 4 accesses on external bus
    → increases access time by factor 4

Implementation FMC

- CPU write to memory
  - Address and data stored in FMC FIFO buffer
  - Avoids wait for slow memory
  - Free system bus for other accesses
- CPU read from memory
  - System bus has to wait until external memory device provides data



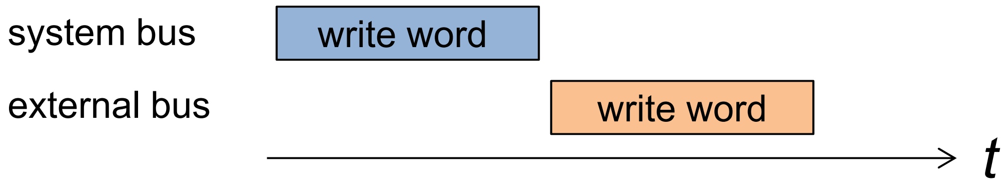
[1] I.e. a memory device with 8 data lines

Flexible Memory Controller

- Word stored in FMC-FIFO
- System bus is released for other accesses
- FMC-FIFO content is transferred to external memory using 1 to 4 bus cycles

Writing a 32-bit Word from System Bus (32 Data Lines)

- to a 32-bit wide external memory (32 data lines)

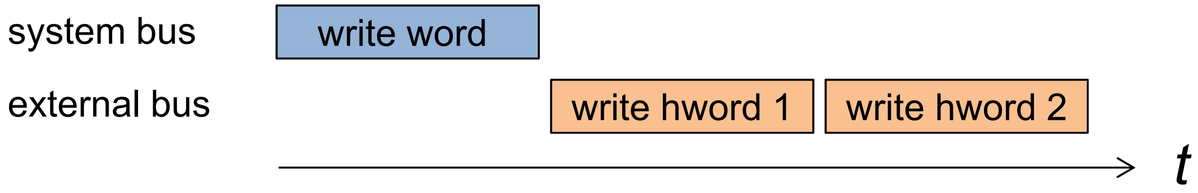


system bus: write word

external bus: write word

t

- to a 16-bit wide external memory (16 data lines)

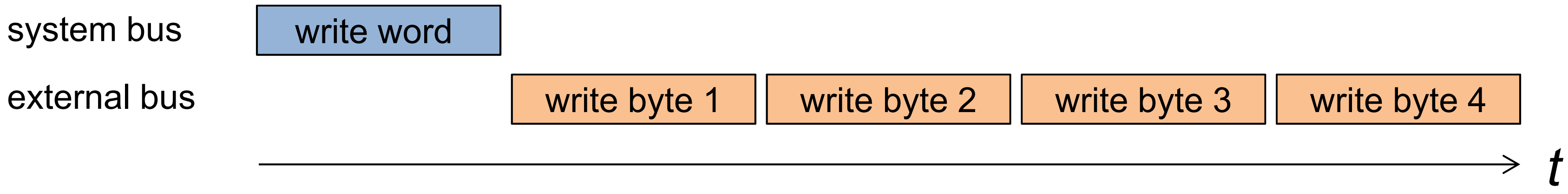


system bus: write word

external bus: write hword 1, write hword 2

t

- to an 8-bit wide external memory (8 data lines)



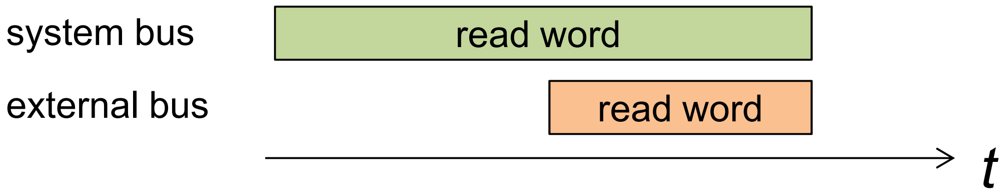
system bus: write word

external bus: write byte 1, write byte 2, write byte 3, write byte 4

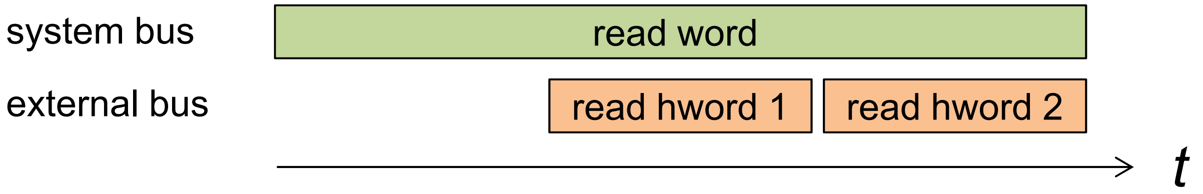
t

Flexible Memory Controller

- **Reading a 32-bit Word from**
  - a 32-bit wide external memory (32 data lines)

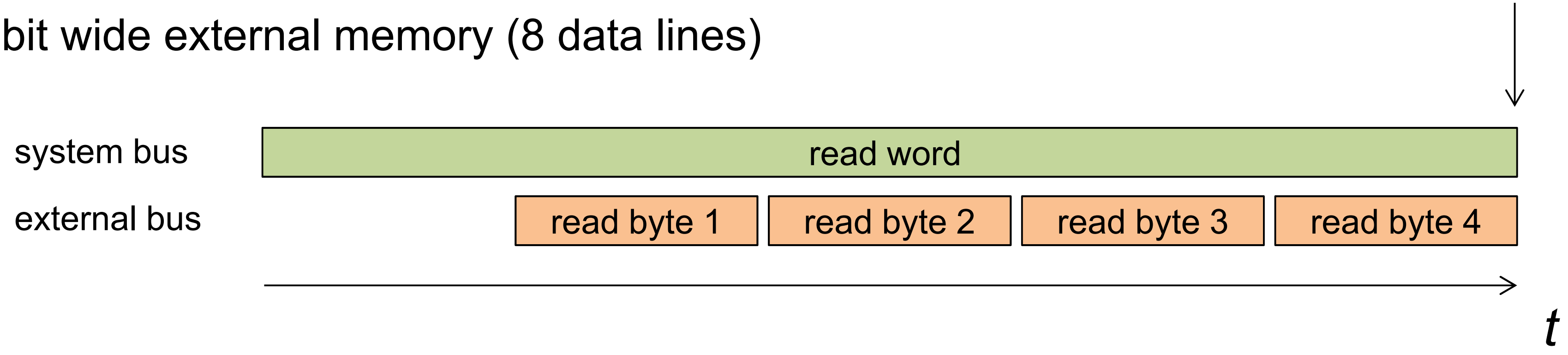


  - a 16-bit wide external memory (16 data lines)



system bus has to wait until all data is available

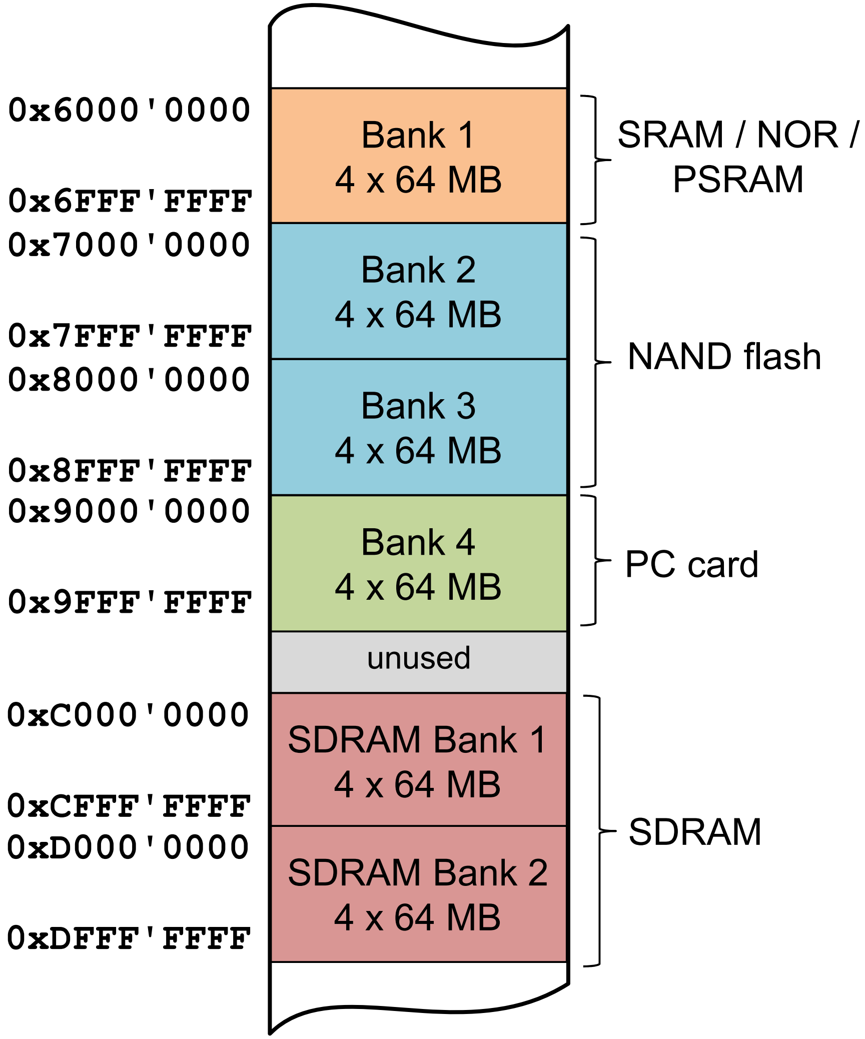
  - an 8-bit wide external memory (8 data lines)



Flexible Memory Controller

- **FMC – Memory Banks**[1)]
  - ST defined address ranges for each type of memory
  - Organized in 6 banks
  - Each bank allows connection of 4 devices
  - Pins are multiplexed
    - Not possible to fully use all the banks simultaneously

Memory banks and their location in memory

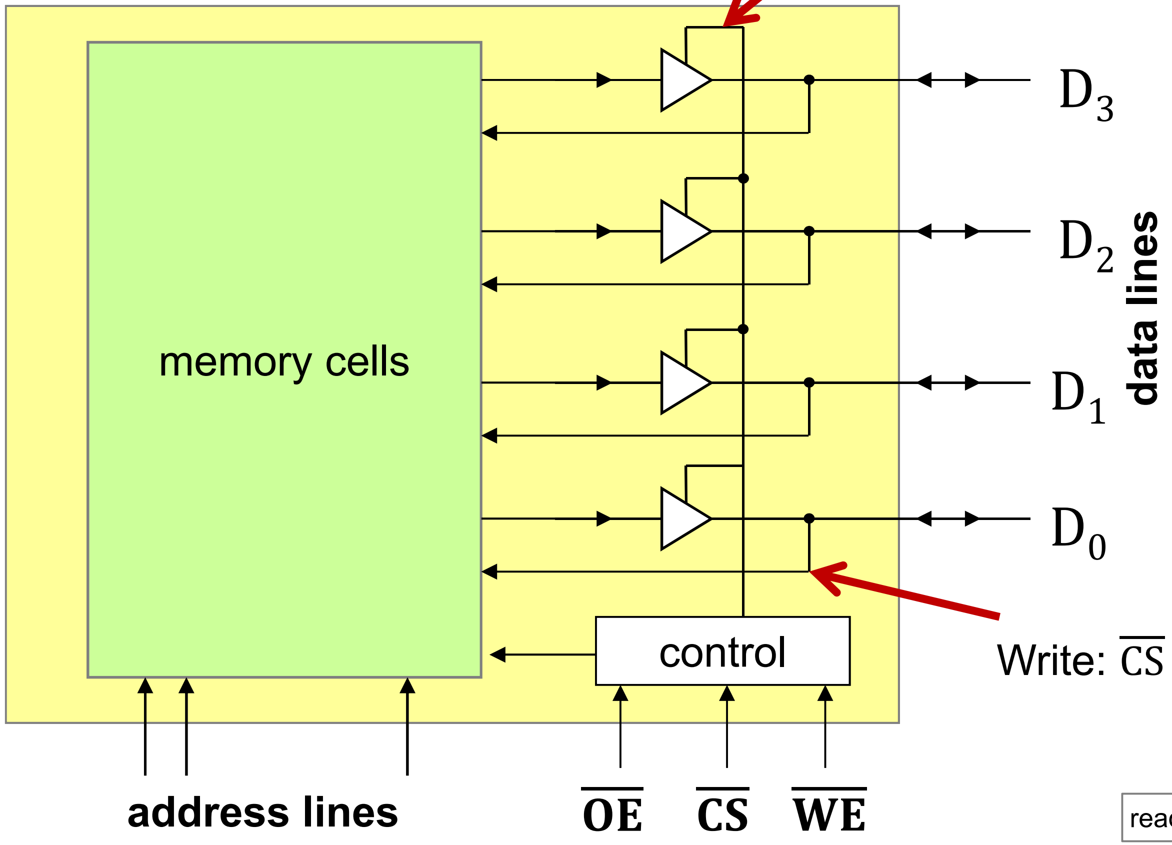


1) An organizational unit of memory. Bank size is architecture dependent

Asynchronous SRAM

- **Asynchronous SRAM Device**
  - No clock input

Read: $\overline{CS}$ = 0 AND $\overline{OE}$ = 0 AND $\overline{WE}$ = 1



Write: $\overline{CS}$ = 0 AND $\overline{WE}$ = 0

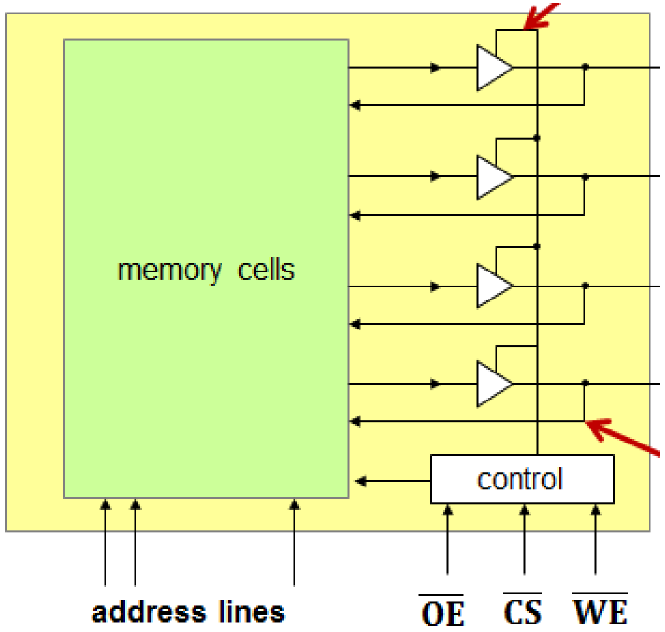
Control signals

$\overline{CS}$	Chip Select usually active-low
$\overline{OE}$	Output Enable usually active-low
$\overline{WE}$	Write enable usually active-low

read and write are from the bus master's perspective

Asynchronous SRAM

- **Asynchronous SRAM Device**
  - I.e. the device does not have a clock signal.



Alternatively the control logic can be represented with a truth table

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	I/O	Function
L	L	H	DATA OUT	Read Data
L	X	L	DATA IN	Write Data
L	H	H	HIGH-Z	Outputs Disabled
H	X	X	HIGH-Z	Deselected

Some memory vendors call the signal $\overline{CE}$ (chip enable) instead of $\overline{CS}$

Asynchronous SRAM

- **Exercise: Example Asynchronous SRAM Device**
  - Use the Data Sheet IDT 71V124SA15TYG
    - Type of memory ?
    - Memory size in bit ?
    - Organization ___K x _ bit ?
    - Number of pins ?
    - Operating voltage ?
    - Which inputs and outputs does the device have ?
    - Truth Table of the control logic ?
    - Access time (read und write) in ns ?
    - Number of accesses per second (in MHz) ?
    - Maximum power [mW] in operating and in full standby ?

Asynchronous SRAM

- **FMC Signals for SRAMs**
  - Prefix 'N' → active-low signal

FMC signal name	I/O	Function
A[25:0]	OUT	Address bus
D[31:0]	INOUT	Data bidirectional bus
NE[4:1]	OUT	Four enable lines [1]
NOE	OUT	Output enable
NWE	OUT	Write enable
NBL[3:0]	OUT	Byte enable

see "Synchronous Bus" in slide set "Microcontroller Basics"

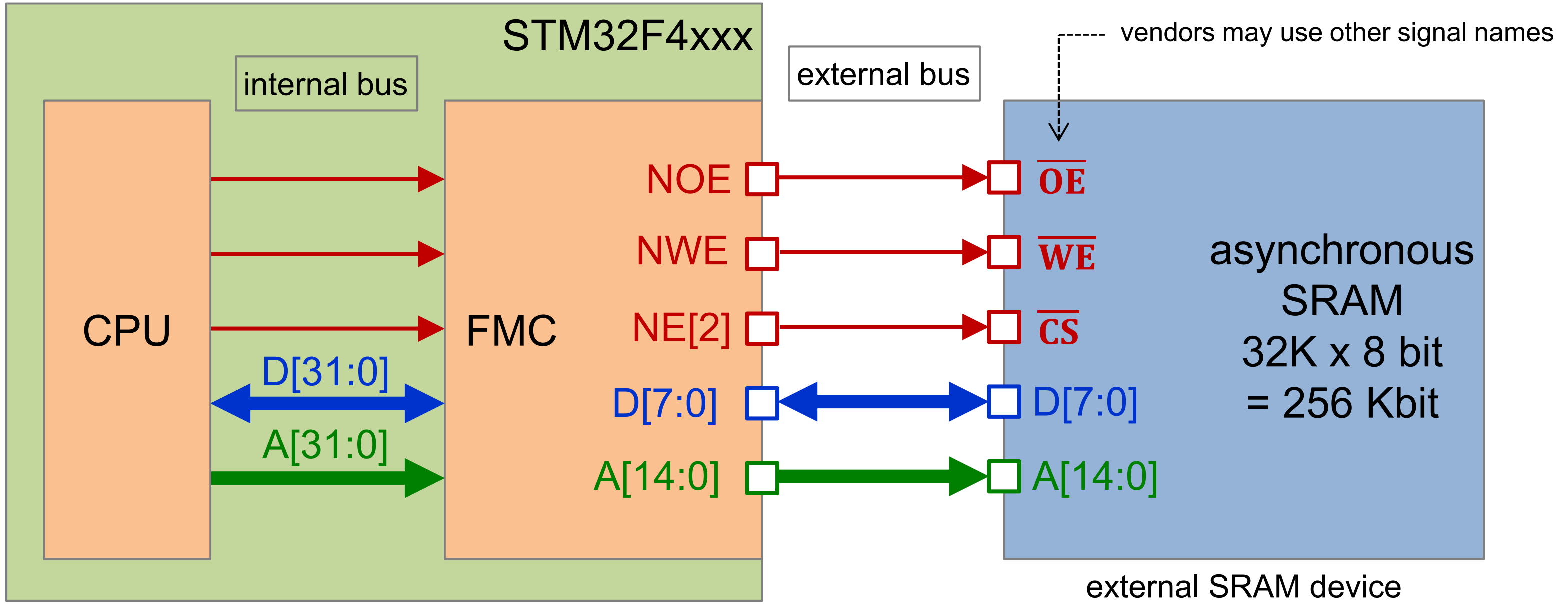
  - Write accesses: NBL[3:0] indicate which bytes shall be updated (see lab)
    - Example 32-bit data bus D[31:0]

► Word access	→ all four bytes	NBL[3:0] = 0000b
► Half-word access	→ two out of four bytes	e.g. NBL[3:0] = 0011b
► Byte access	→ one out of four bytes	e.g. NBL[3:0] = 1011b

[1] ST Microelectronics starts numbering at 1 and not at 0

Asynchronous SRAM

- **Example 32K x 8 bit SRAM**
  - Connecting an external 8-bit asynchronous SRAM device



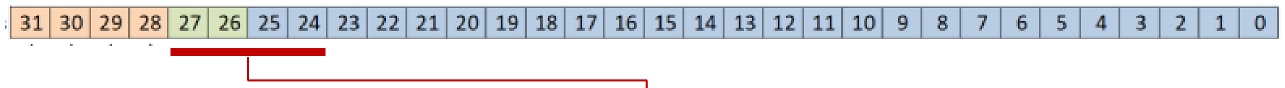
Asynchronous SRAM

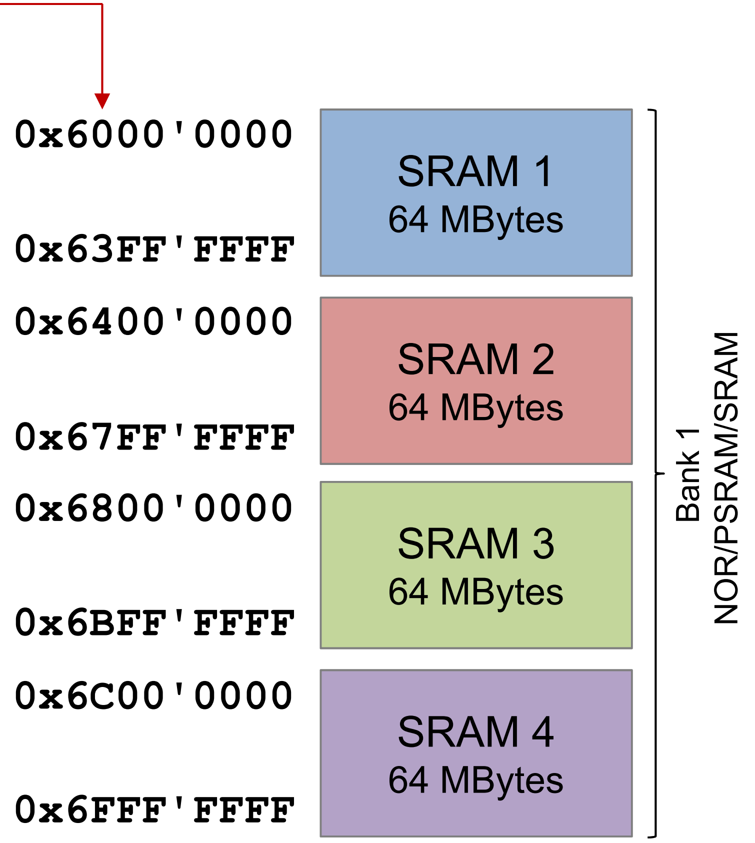
FMC – SRAM (Bank 1)

- Select one out of four SRAM devices
  - Address bits 27:26 → Encoded in signals NE[4:1]

A[27:26]	Enable	Memory Device
00	NE[1]	SRAM 1
01	NE[2]	SRAM 2
10	NE[3]	SRAM 3
11	NE[4]	SRAM 4

- Data bus configured in control registers
  - Example
    - ▶ SRAM1 as 32-bit → D[31:0]
    - ▶ SRAM2 as 8-bit → D[7:0] only
    - ▶ SRAM3 as 16-bit → D[15:0] only
    - ▶ SRAM4 as 32-bit → D[31:0]

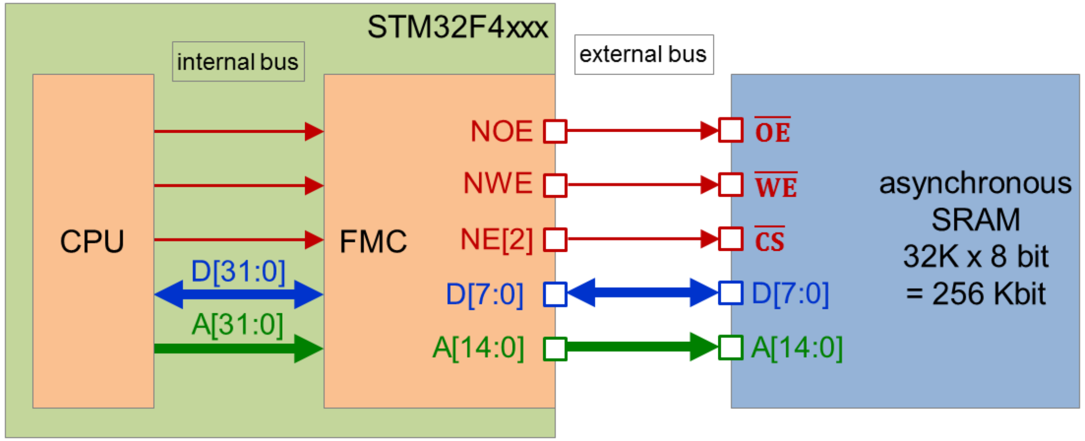


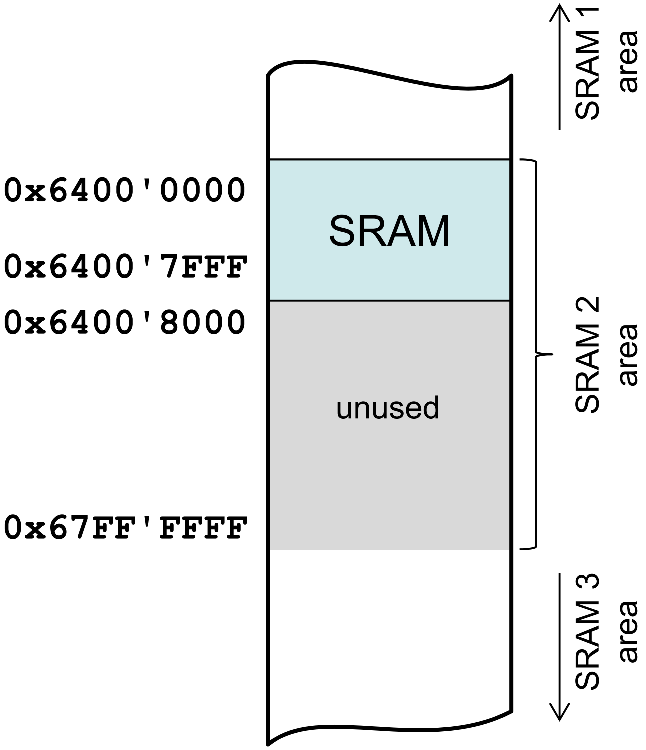


Often the blocks are only used partially

Asynchronous SRAM

- **Example (revisited)**
  - Memory map of our previous 32K x 8 bit SRAM



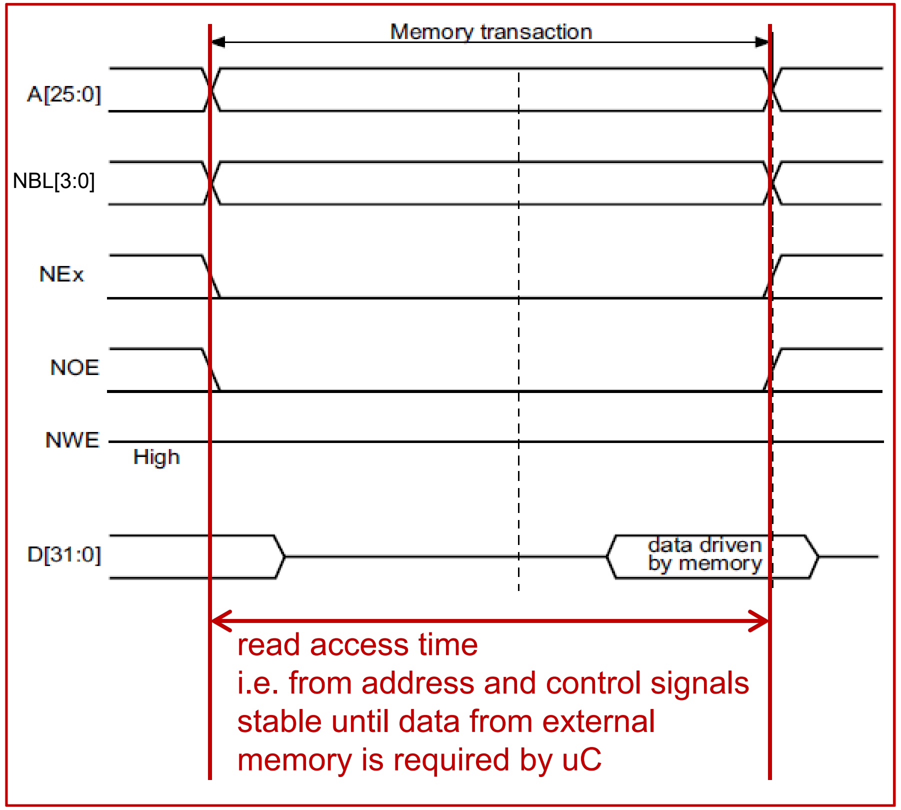


Asynchronous SRAM

- **Timing on External Bus**
  - As seen from the microcontroller

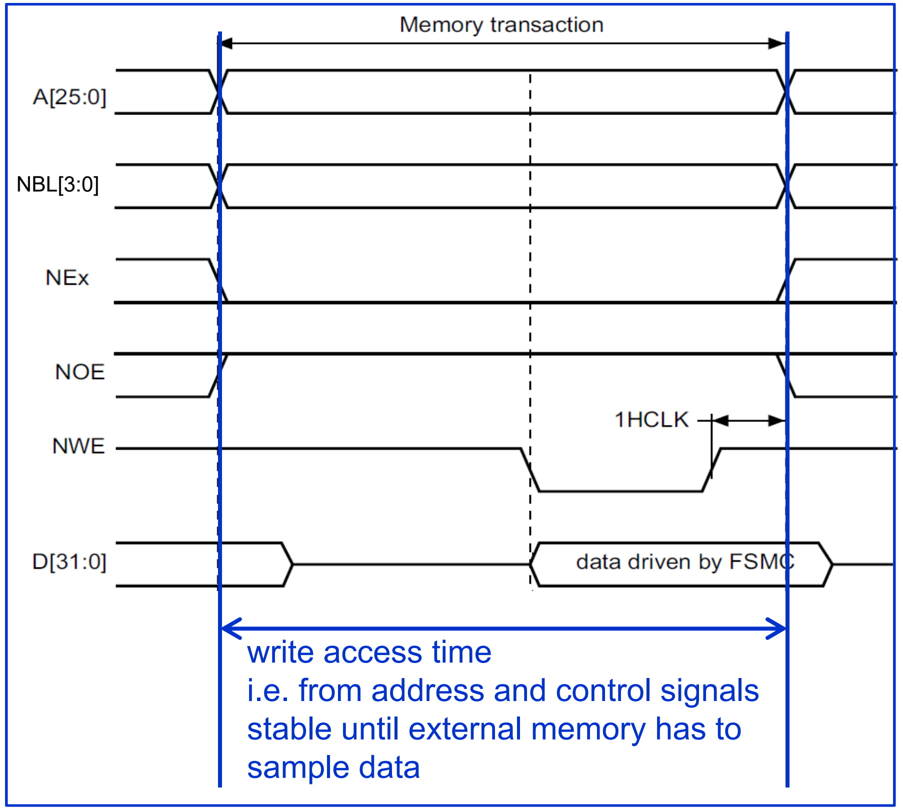
Figures from STM32F4xxx reference manual
p. 1591, chapter 37, Flexible Memory Controller

Read Access



read access time
i.e. from address and control signals stable until data from external memory is required by uC

Write Access

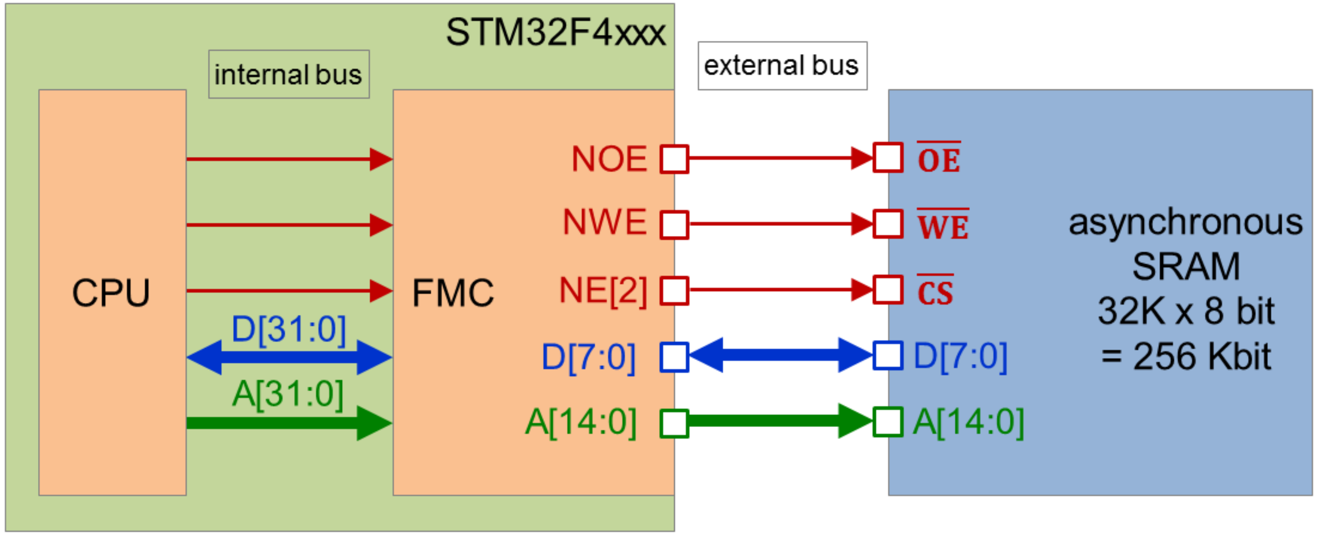


write access time
i.e. from address and control signals stable until external memory has to sample data

HCLK is the clock period of the CPU and the internal data bus

Asynchronous SRAM

- **Configuration of FMC**
  - Location of FMC control registers
    - 0xA000'0000 – 0xA000'0FFF
  - Configure FMC according to SRAM datasheet
    - Data bus size → 8-bit, 16-bit, 32-bit
    - Access times
    - and others



The FMC Registers allow configuration for many different memory types. However we only cover a few selected parameters for asynchronous SRAM.

Asynchronous SRAM

- **Configuring the FMC for SRAM**

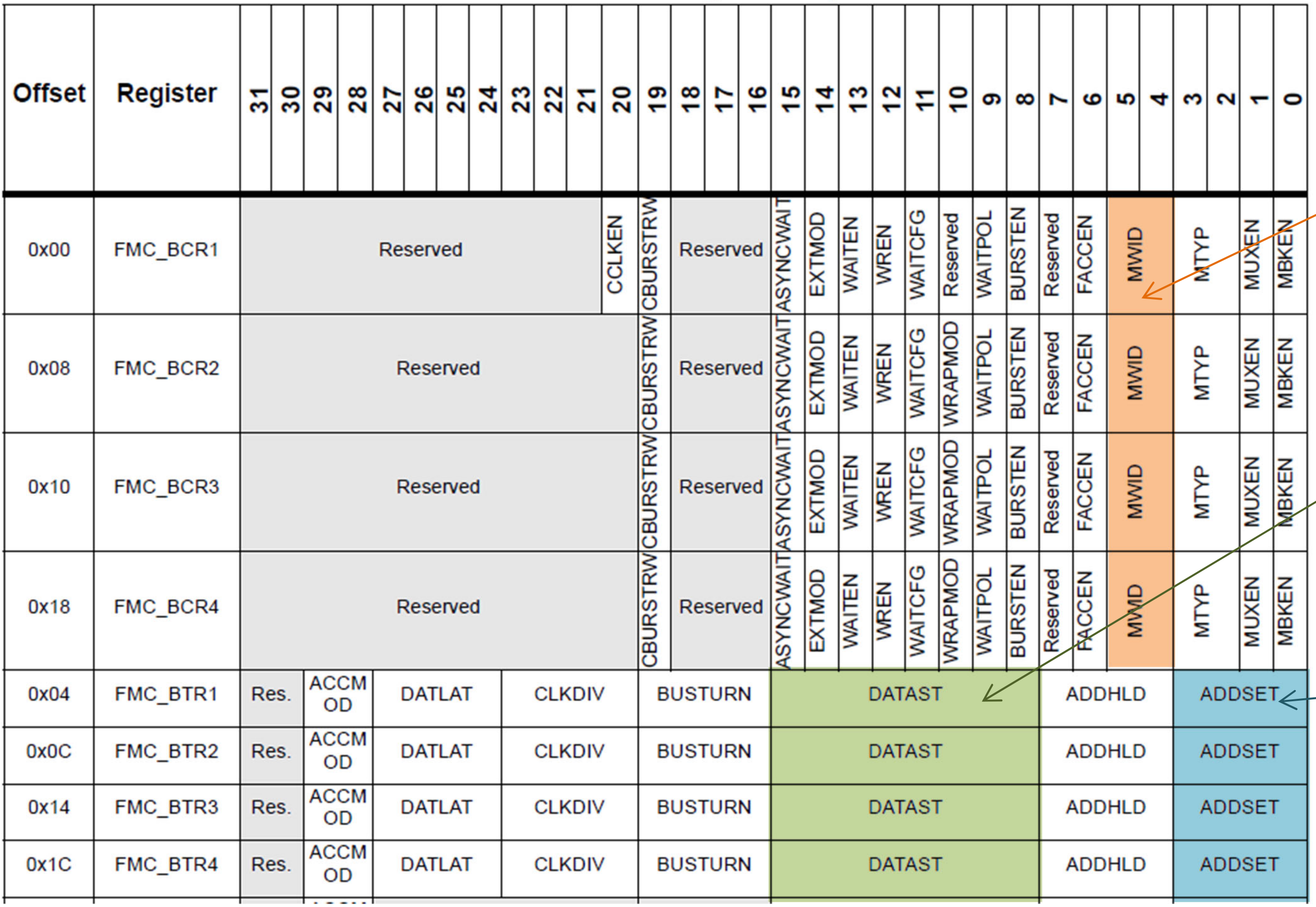
Control register for SRAM1 → @ 0xA000'0000

Control register for SRAM2 → @ 0xA000'0008

Control register for SRAM3 → @ 0xA000'0010

Timing register for SRAM1 → @ 0xA000'0004

Table 291. FMC register map



Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FMC_BCR1	Reserved											CCLKEN	CBURSTRW	Reserved			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Reserved	WAITPOL	BURSTEN	Reserved	FACCEN	MWID		MTYP		MUXEN	MBKEN
0x08	FMC_BCR2	Reserved												CBURSTRW	Reserved			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID		MTYP		MUXEN	MBKEN
0x10	FMC_BCR3	Reserved												CBURSTRW	Reserved			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID		MTYP		MUXEN	MBKEN
0x18	FMC_BCR4	Reserved												CBURSTRW	Reserved			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID		MTYP		MUXEN	MBKEN
0x04	FMC_BTR1	Res.		ACCMOD		DATLAT				CLKDIV				BUSTURN				DATAST								ADDHLD				ADDSET			
0x0C	FMC_BTR2	Res.		ACCMOD		DATLAT				CLKDIV				BUSTURN				DATAST								ADDHLD				ADDSET			
0x14	FMC_BTR3	Res.		ACCMOD		DATLAT				CLKDIV				BUSTURN				DATAST								ADDHLD				ADDSET			
0x1C	FMC_BTR4	Res.		ACCMOD		DATLAT				CLKDIV				BUSTURN				DATAST								ADDHLD				ADDSET			

00	8 bits
01	16 bits
10	32 bits
11	reserved

Use between 1 and 255 HCLK cycles during data phase.

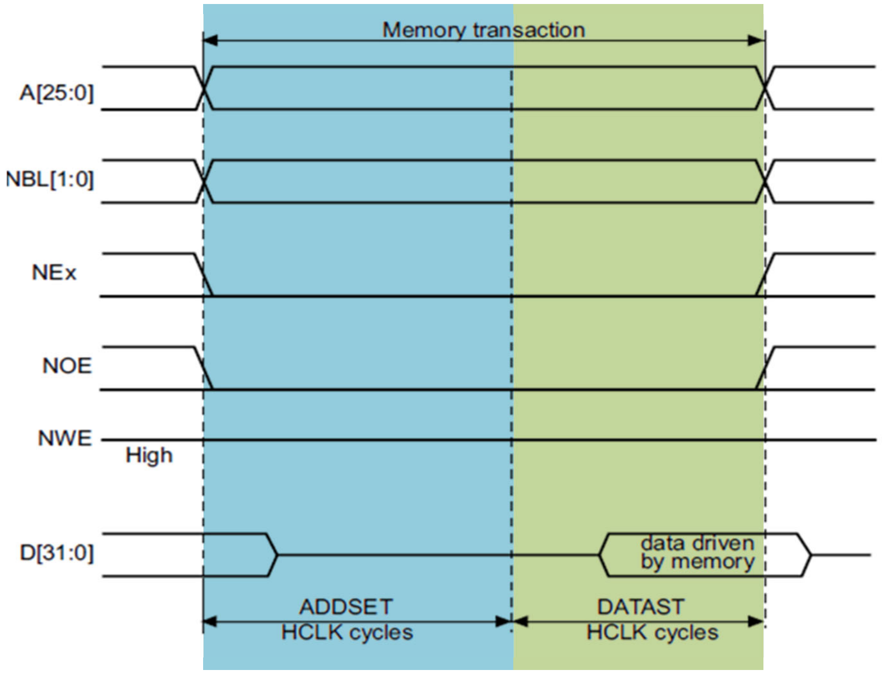
Use between 1 and 15 HCLK cycles during address phase

Asynchronous SRAM

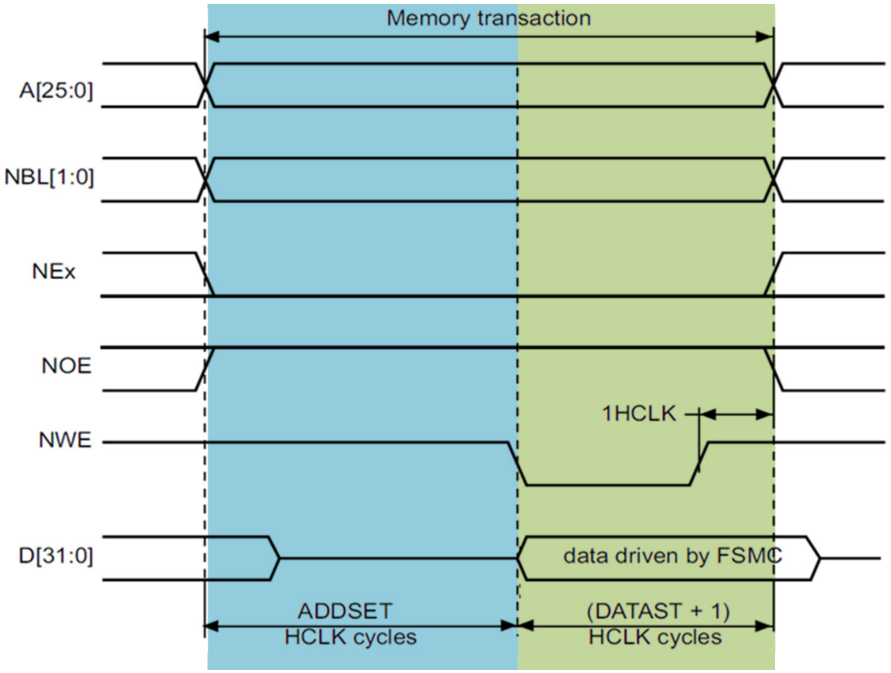
ADDSET and DATAST

- Configuring length of access cycles
- HCLK programmed to 84 MHz during start-up of CT-Board
  - HCLK = Frequency of CPU and internal bus

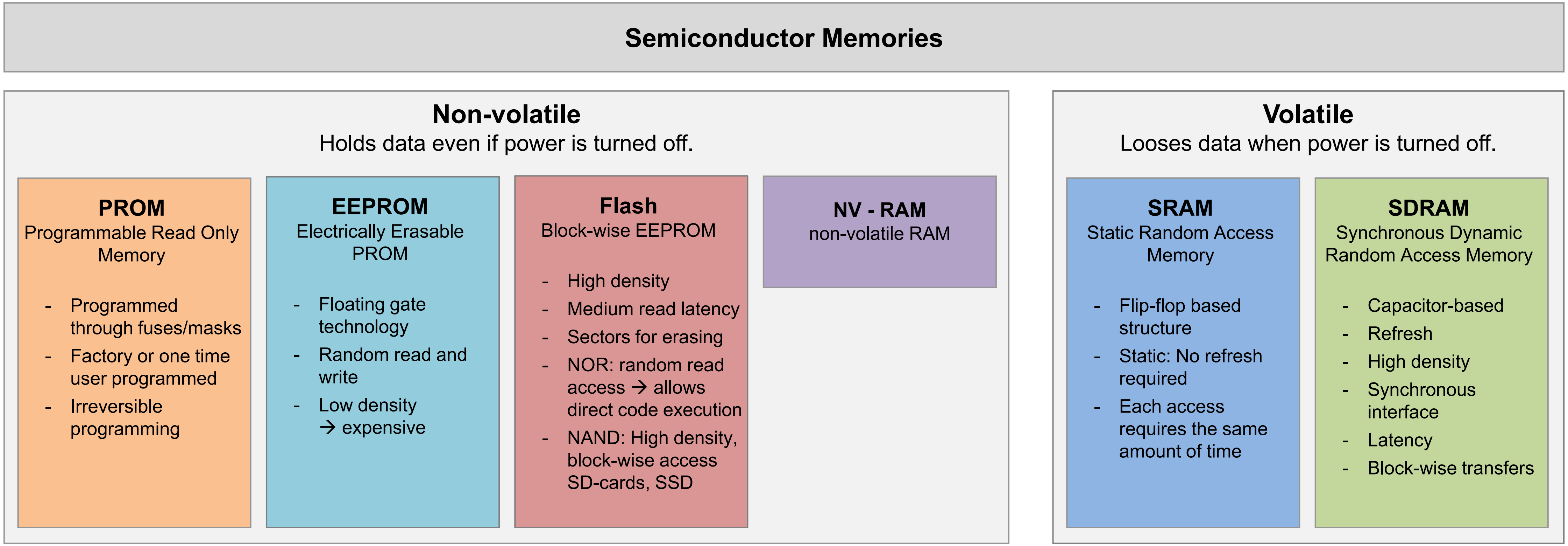
Read Access



Write Access



Conclusions



Semiconductor Memories

Non-volatile
Holds data even if power is turned off.

PROM
Programmable Read Only Memory

- Programmed through fuses/masks
- Factory or one time user programmed
- Irreversible programming

EEPROM
Electrically Erasable PROM

- Floating gate technology
- Random read and write
- Low density → expensive

Flash
Block-wise EEPROM

- High density
- Medium read latency
- Sectors for erasing
- NOR: random read access → allows direct code execution
- NAND: High density, block-wise access SD-cards, SSD

NV - RAM
non-volatile RAM

Volatile
Looses data when power is turned off.

SRAM
Static Random Access Memory

- Flip-flop based structure
- Static: No refresh required
- Each access requires the same amount of time

SDRAM
Synchronous Dynamic Random Access Memory

- Capacitor-based
- Refresh
- High density
- Synchronous interface
- Latency
- Block-wise transfers

- **Flexible Memory Controller STM32**
  - Configurable bridge to connect external memories → e.g. asynchronous SRAM, NOR flashes, etc.

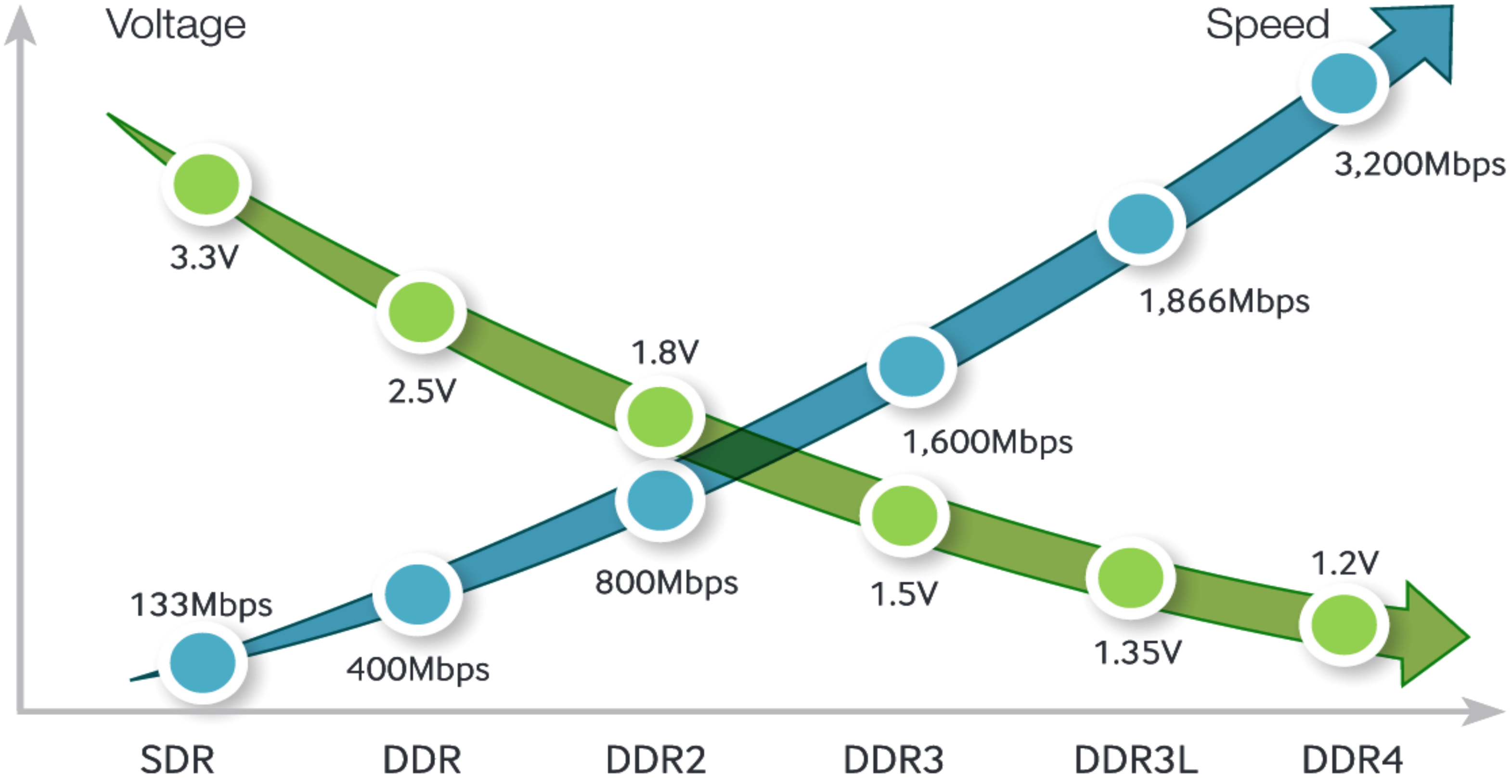
For Information Only

TRENDS AND FIGURES

SDRAM – Synchronous Dynamic RAM

SDR	Single Data Rate
DDR	Dual Data Rate uses rising and falling clock edge

- **Voltage and Speed**
  - Different generations of SDRAM



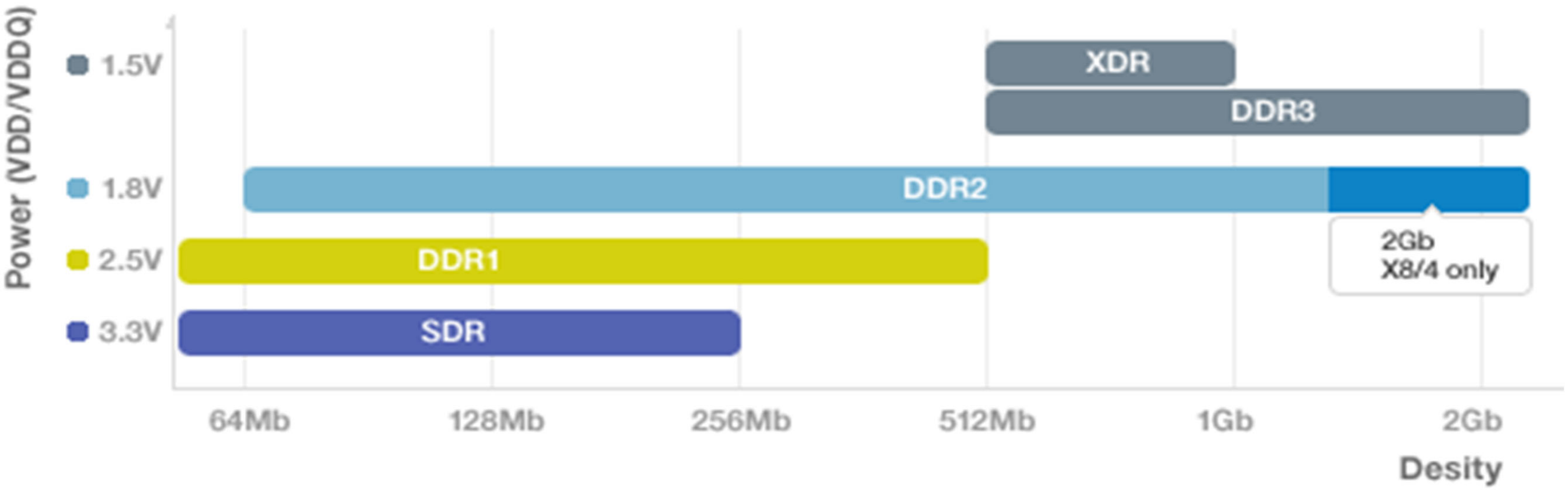
source: Samsung

SDRAM – Synchronous Dynamic RAM

- **Densities**

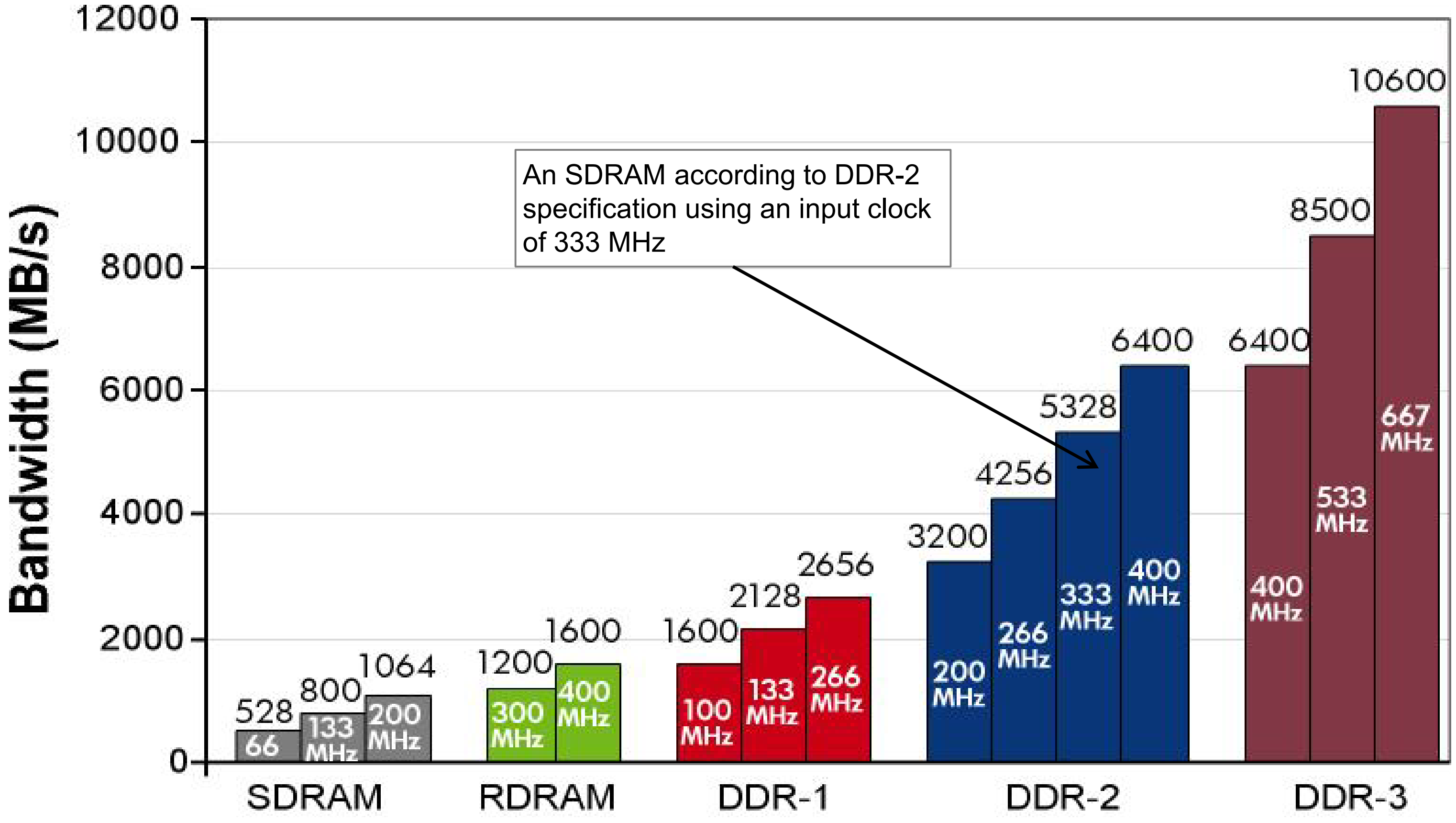
DRAM Density & Power Supply for Consumer Applications in 2009~2010

SAMSUNG supports 1.5V, thus meeting customer demand in a wide range of applications.



SDRAM – Synchronous Dynamic RAM

- **Peak Bandwidth**

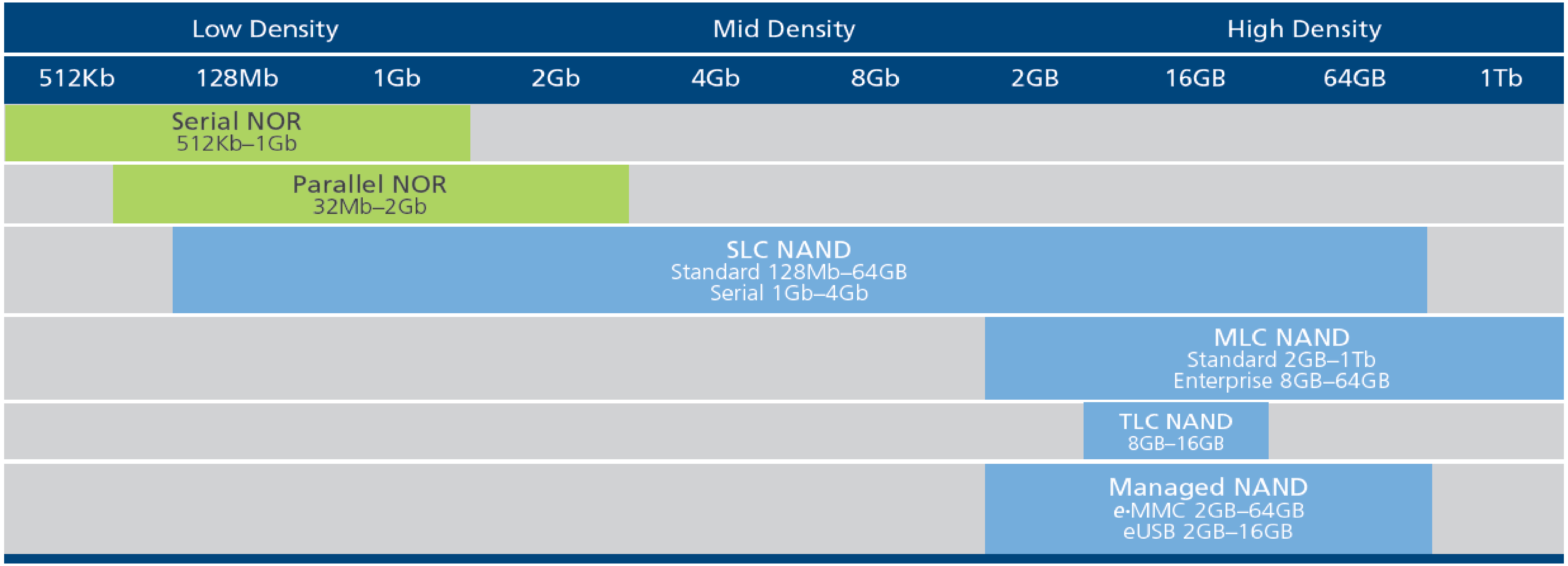


An SDRAM according to DDR-2 specification using an input clock of 333 MHz

Source: HP: Memory technology evolution: an overview of system memory technologies

Non-volatile Memory – Flash

- **Flash Densities**



Low Density

Mid Density

High Density

512Kb

128Mb

1Gb

2Gb

4Gb

8Gb

2GB

16GB

64GB

1Tb

Serial NOR
512Kb–1Gb

Parallel NOR
32Mb–2Gb

SLC NAND
Standard 128Mb–64GB
Serial 1Gb–4Gb

MLC NAND
Standard 2GB–1Tb
Enterprise 8GB–64GB

TLC NAND
8GB–16GB

Managed NAND
e·MMC 2GB–64GB
eUSB 2GB–16GB

Source: Micron Technology, Inc., 2014

Managed NAND includes a controller for tasks like error correction.