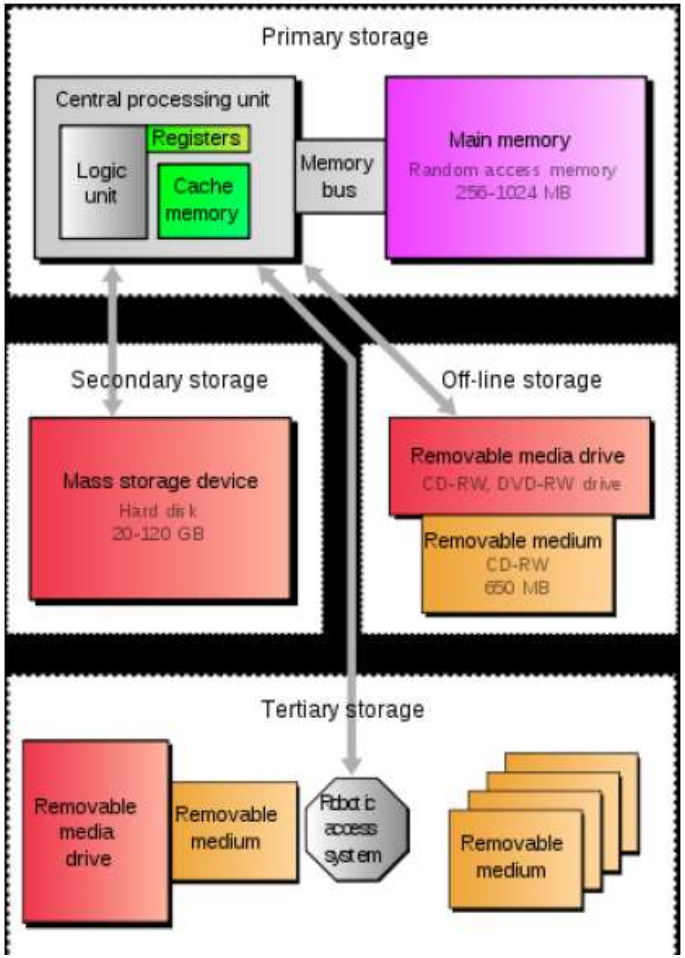
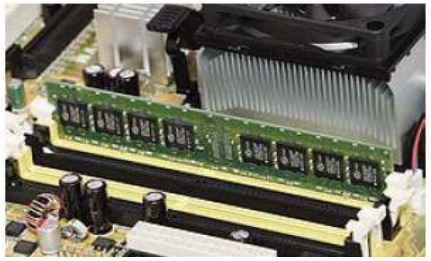
Memory technologies and Embedded Memories

Memory: Basics (2)

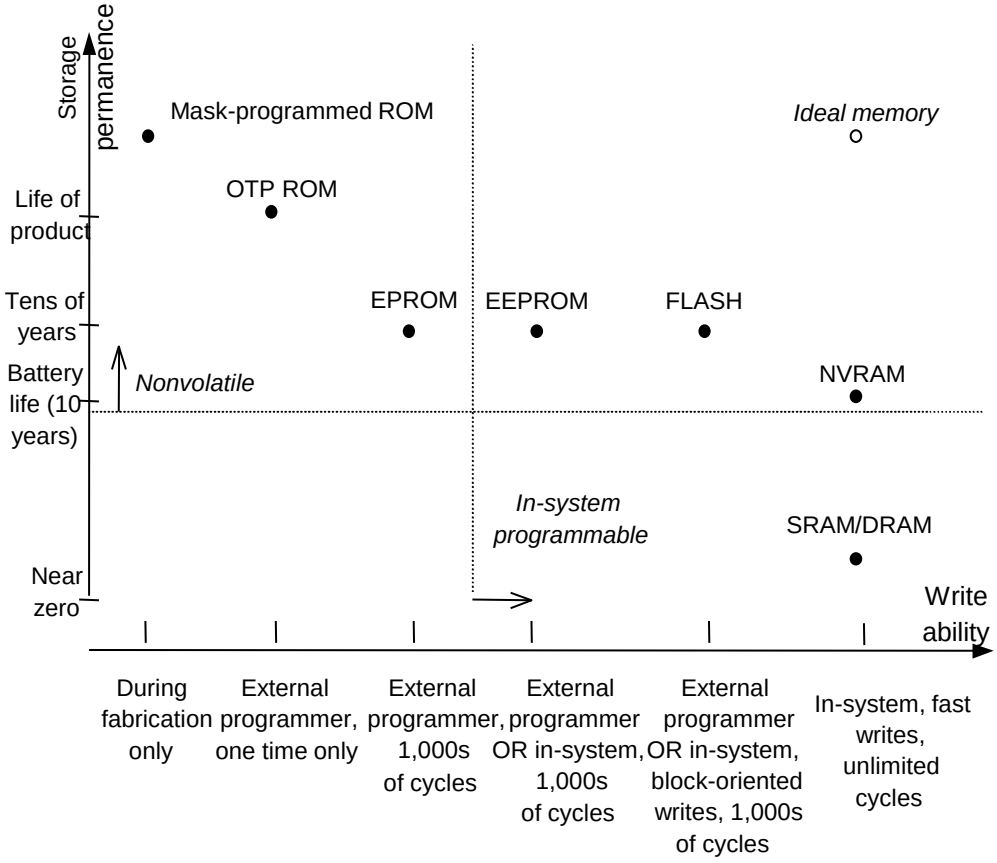




- Historically RAM and Hard disk were called primary and secondary storages respectively
- Hierarchy of computer storage:
  - Primary
  - Secondary
  - Tertiary

Write Ability/ Storage Permanence

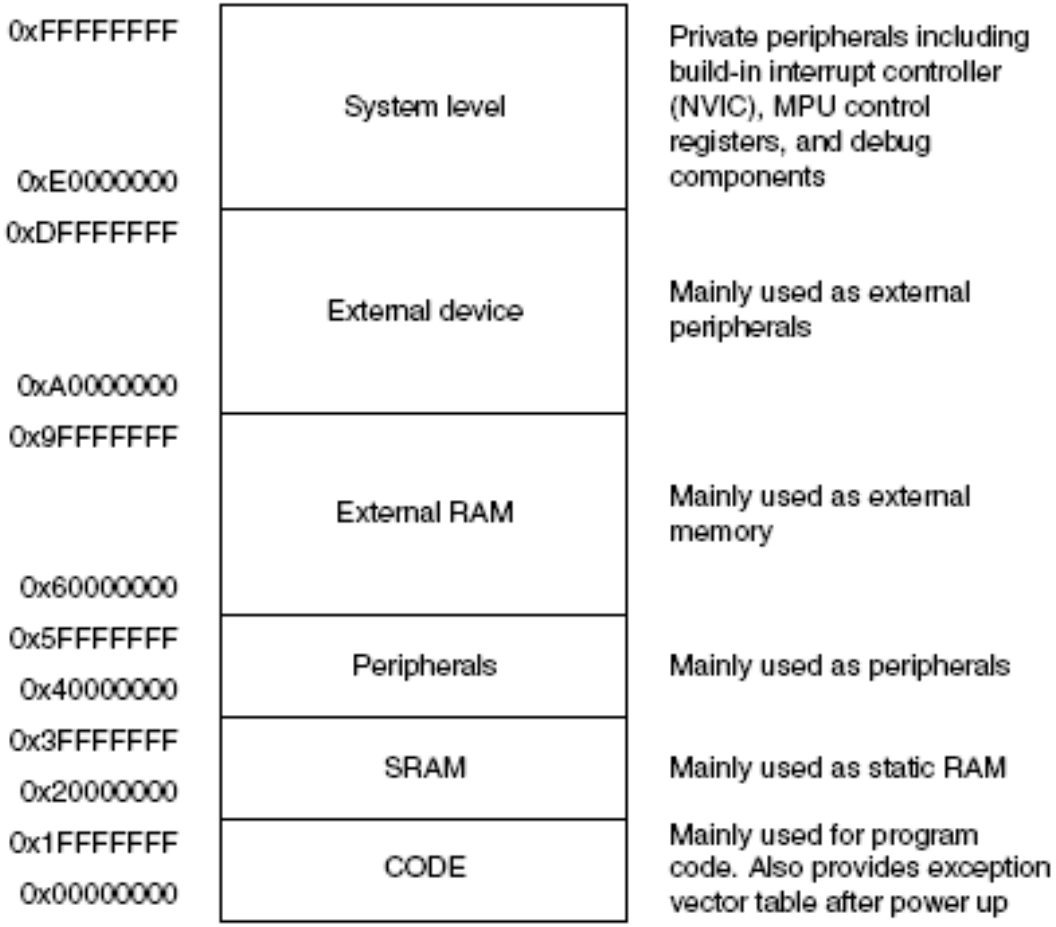
- Traditional ROM/RAM distinctions
  - ROM
    - read only, bits stored without power
  - RAM
    - read and write, lose stored bits without power
- Traditional distinctions blurred
  - Advanced ROMs can be written to
    - e.g., EEPROM
  - Advanced RAMs can hold bits without power
    - e.g., NVRAM
- Write ability
  - Manner and speed a memory can be written
- Storage permanence
  - ability of memory to hold stored bits after they are written



Write ability and storage permanence of memories, showing relative degrees along each axis (not to scale).

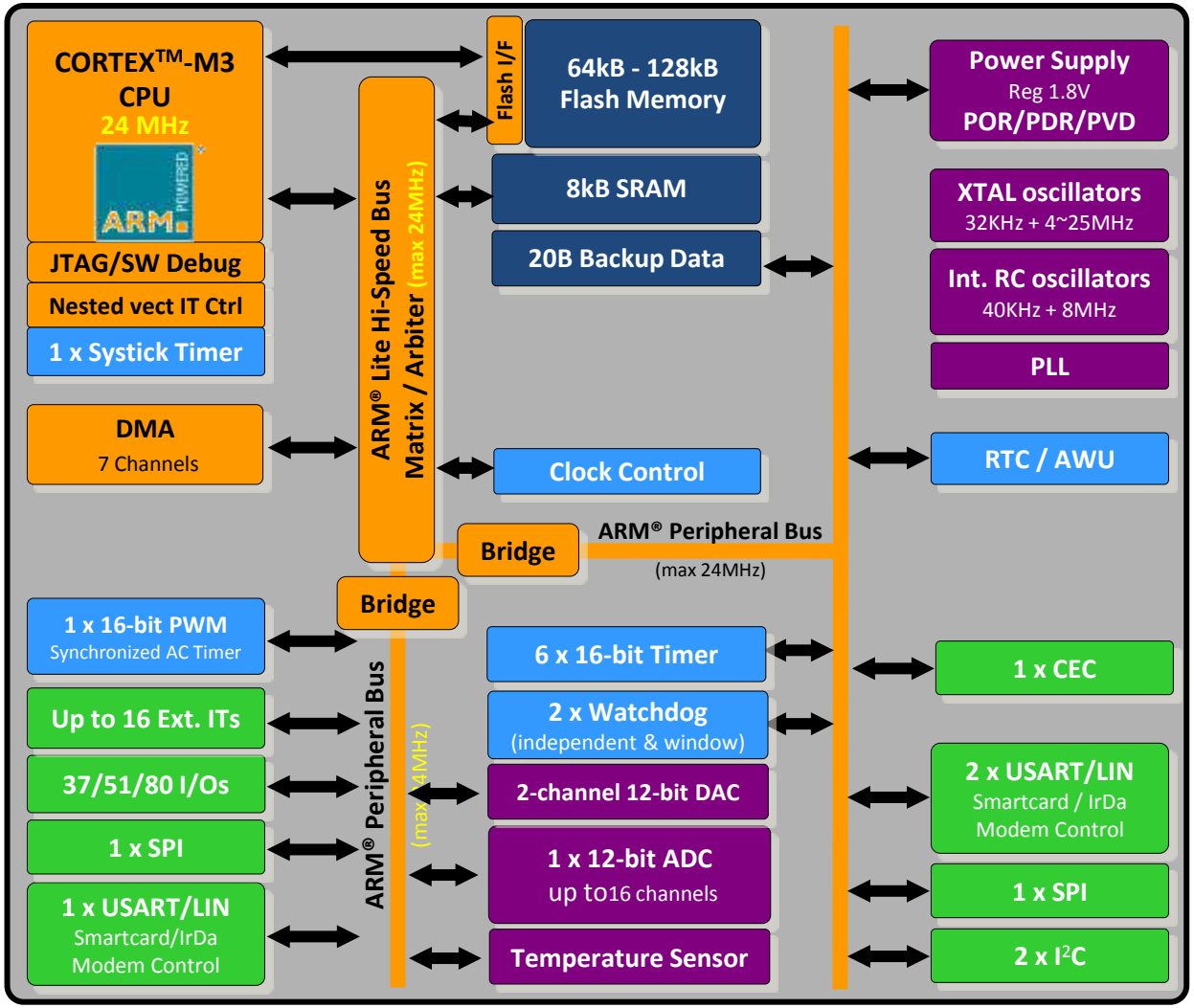
Memory map

- Statically defined memory map (faster addr decoding) 4GB of address psace



STM32 Value line 64K-128KBytes System Diagram

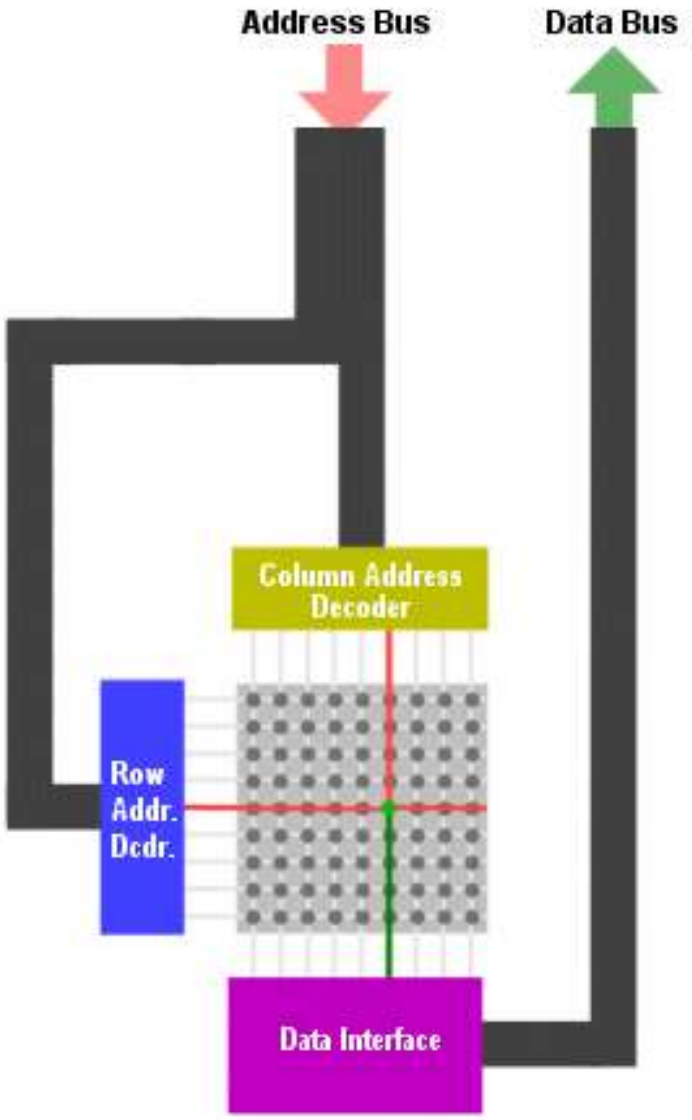
- **Core and operating conditions**
  - ARM® Cortex™-M3
  - 1.25 DMIPS/MHz up to 24 MHz
  - 2.0 V to 3.6 V range
  - -40 to +105 °C
- **Rich connectivity**
  - 8 communications peripherals
- **Advanced analog**
  - 12-bit1.2 µs conversion time ADC
  - Dual channel 12-bit DAC
- **Enhanced control**
  - 16-bit motor control timer
  - 6x 16-bit PWM timers
- **LQFP48, LQFP/BGA64, LQFP100**



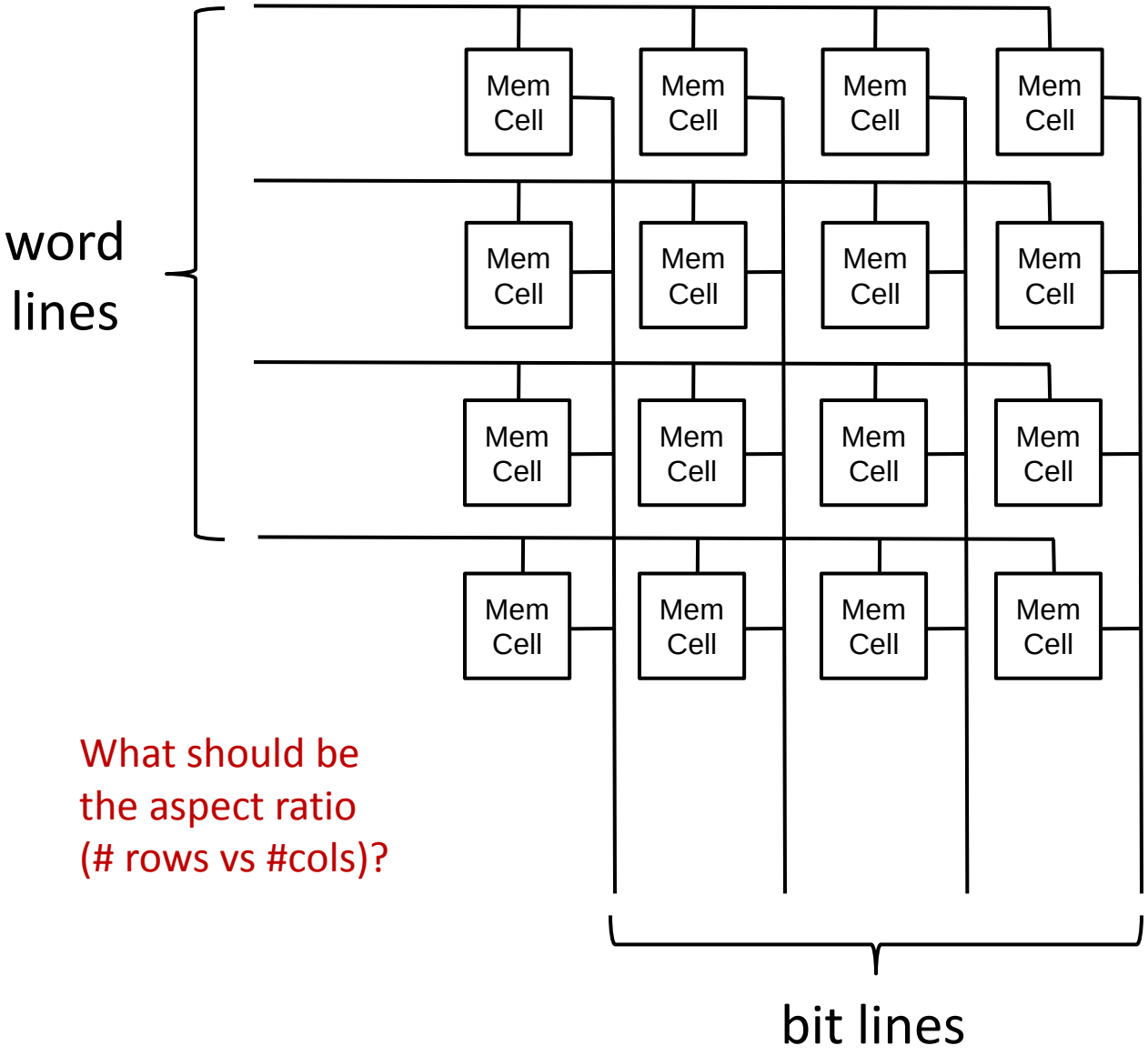
Solid-state Memories (refresher)

Storage Basics

- Just because the CPU sees RAM as one long, thin line of bytes doesn't mean that it's actually laid out that way
- Real RAM chips don't store whole bytes, but rather they store individual bits in a grid, which you can address one bit at a time
- Types of memory
  - Non Volatile → ROM/EPROM/FLASH
  - Volatile → SRAM, DRAM



Internal organization



Different memory types (e.g. SRAM vs DRAM) are distinguished by the technology used to implement the memory cell, e.g.:

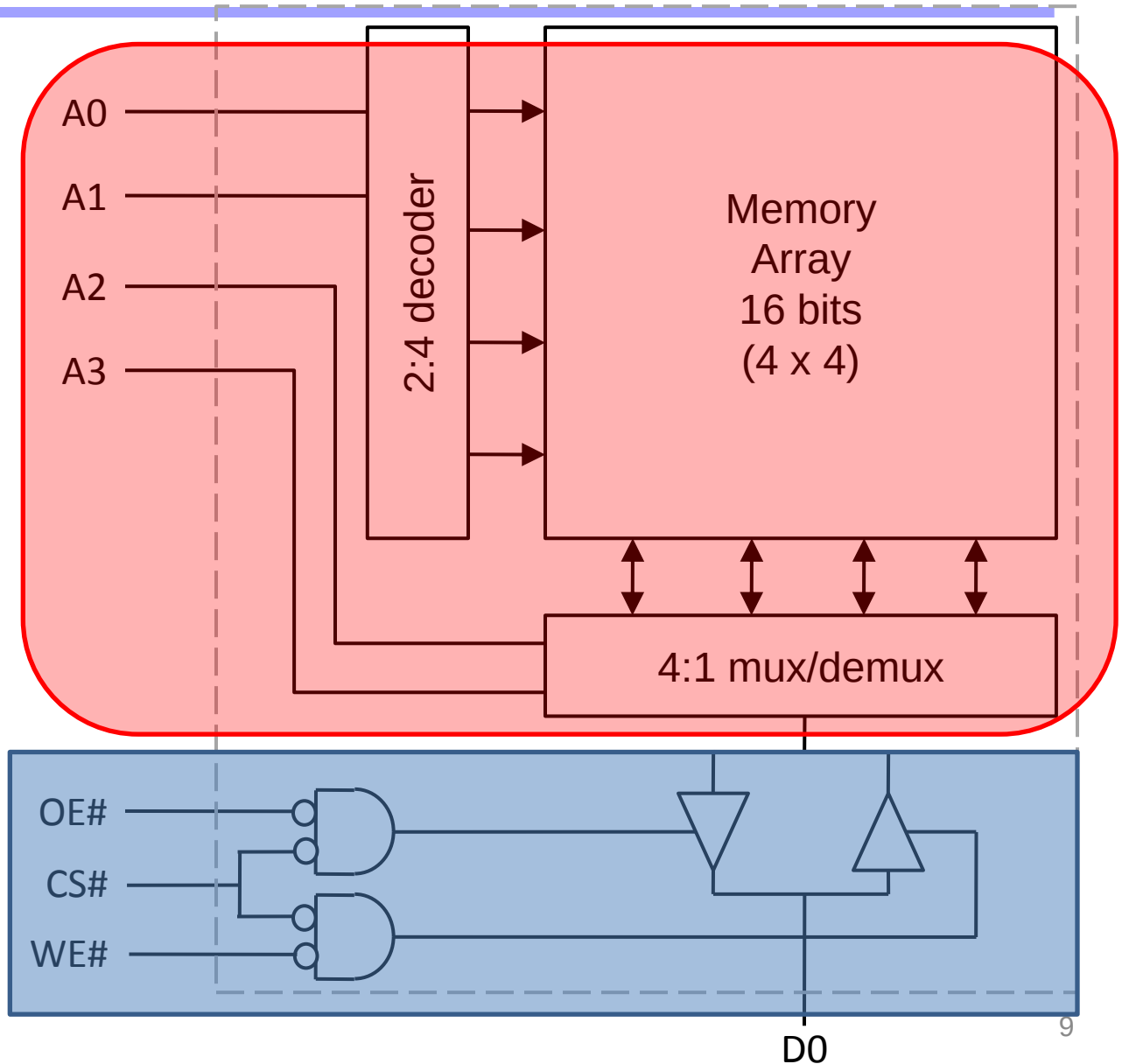
- SRAM: 6T
- DRAM: 1T/1C

What should be the aspect ratio (# rows vs #cols)?

Internal organization

Decoders & Muxes

Data interface



Physical (on-chip) memory configuration

- Physical configurations are typically square
- Square minimizes length of (word line + bit line)
- Shorter length means
  - Shorter propagation time
  - Faster data access
  - Smaller t_{rc} (read cycle time)

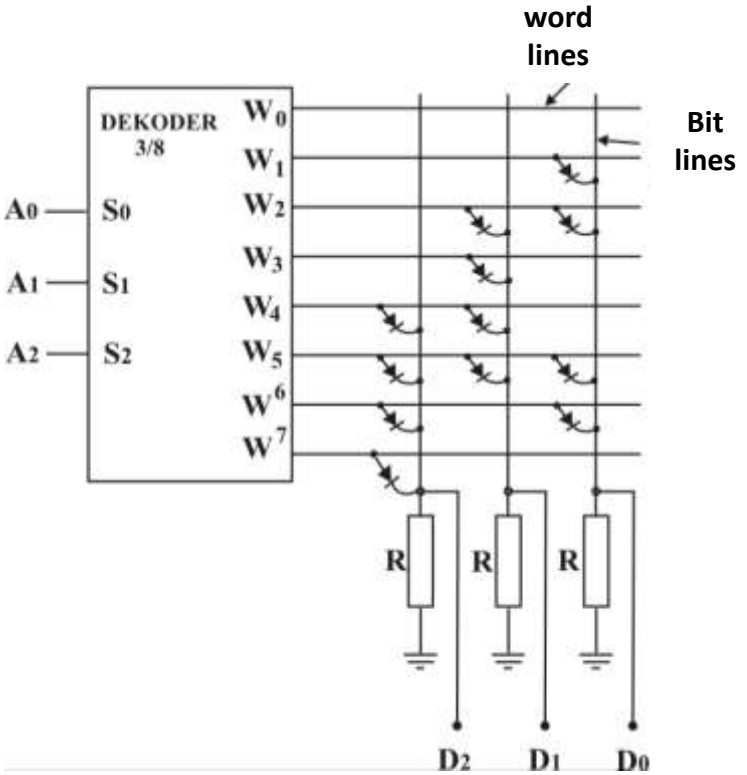
- Exercise: Assume n^2 memory cells configured as
  - n-by-n square array. What is the worst case delay?
  - n^2-by-1 rectangular. What is the worst case delay?
- Exercise: Does wire length dominate access time?
  - Assume propagation speed on chip is 2/3 c (2x10^8 m/s)
  - Assume 1Mbit array is 1 cm x 1 cm

Logical (external) memory configuration

- External configurations are tall and narrow
  - More address lines (12 to 20+, typically)
  - Fewer data lines (8 or 16, typically)
- The narrower the configuration
  - The greater the pin efficiency
  - Adding one address pin cuts data pins in half
  - The easier the data bus routing
- Many external configurations for given capacity
  - 64 Kb = 64K x 1 (16 A + 1 D = 17 pins)
  - 64 Kb = 32K x 2 (15 A + 2 D = 17 pins)
  - 64 Kb = 16K x 4 (14 A + 4 D = 18 pins)
  - 64 Kb = 8K x 8 (13 A + 8 D = 21 pins)
  - 64 Kb = 4K x 16 (12 A + 16 D = 28 pins)
  - 64 Kb = 2K x 32 (11 A + 32 D = 43 pins)

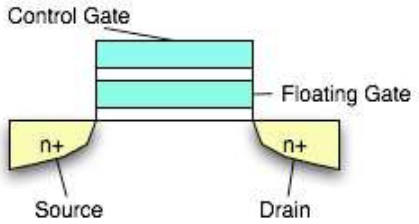
NV: Mask ROM

- The “simplest” memory technology
- Presence/absence of diode at each cell denote value
- Pattern of diodes defined by mask used in fab process
- Contents are fixed when chip is made; cannot be changed
- High upfront setup costs (mask costs)
- Small recurring marginal costs
- Good for applications where
  - Cost sensitivity drives design
  - Upgrading contents not an issue
  - e.g. boot ROM, CPU microcode
- Exercise:
  - What “value” does a diode encode?
  - What are the contents:
    - Where A<2:0> = 101?
    - Where A<2:0> = 110?



NV: EPROM

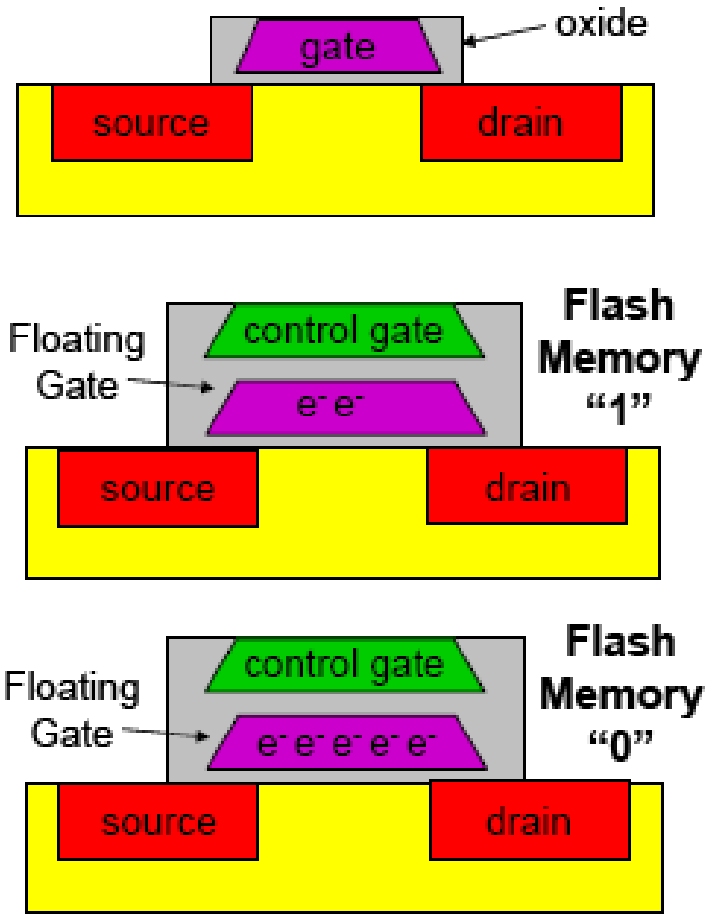
- Erasable Programmable Read-Only Memory
- Constructed from floating gate FETs
  - Charge trapped on the FG erases cell
  - High voltage (13V +) applied to the control gate
    - "Writes" the cell with a 0
    - Allows FG charge to be dissipated
- Erasing means changing form 0 → 1
  - Uses UV light (not electrically!)
  - Electrons are trapped on a floating gate
- Writing means changing from 1 → 0
- Erase unit is the whole device
- Retains data for 10-20 years
- Not used much these days
- Costly because
  - Use of quartz window (UV transparent)
  - Use of ceramic package
- PROM (or OTP) is same, just w/o window



NV: Flash Memory

- Electrically erasable (like EEPROM, unlike EPROM)
- Used in many reprogrammable systems these days
- Erase size is block (not word); can't do byte modifications
- Erase circuitry moved out of cells to periphery
  - Smaller size
  - Better density
  - Lower cost
- Reads are like standard RAM
- Can "write" bits/words (actually, change from 1 → 0)
  - Write cycle is O(microseconds)
  - Slower then RAM but faster than EEPROM
  - To (re)write from 0 → 1, must explicitly erase entire block
    - Erase is time consuming O(milliseconds to seconds)
- Floating gate technology
  - Erase/write cycles are limited (10K to 100K, typically)

Flash Memory - Technology Basics



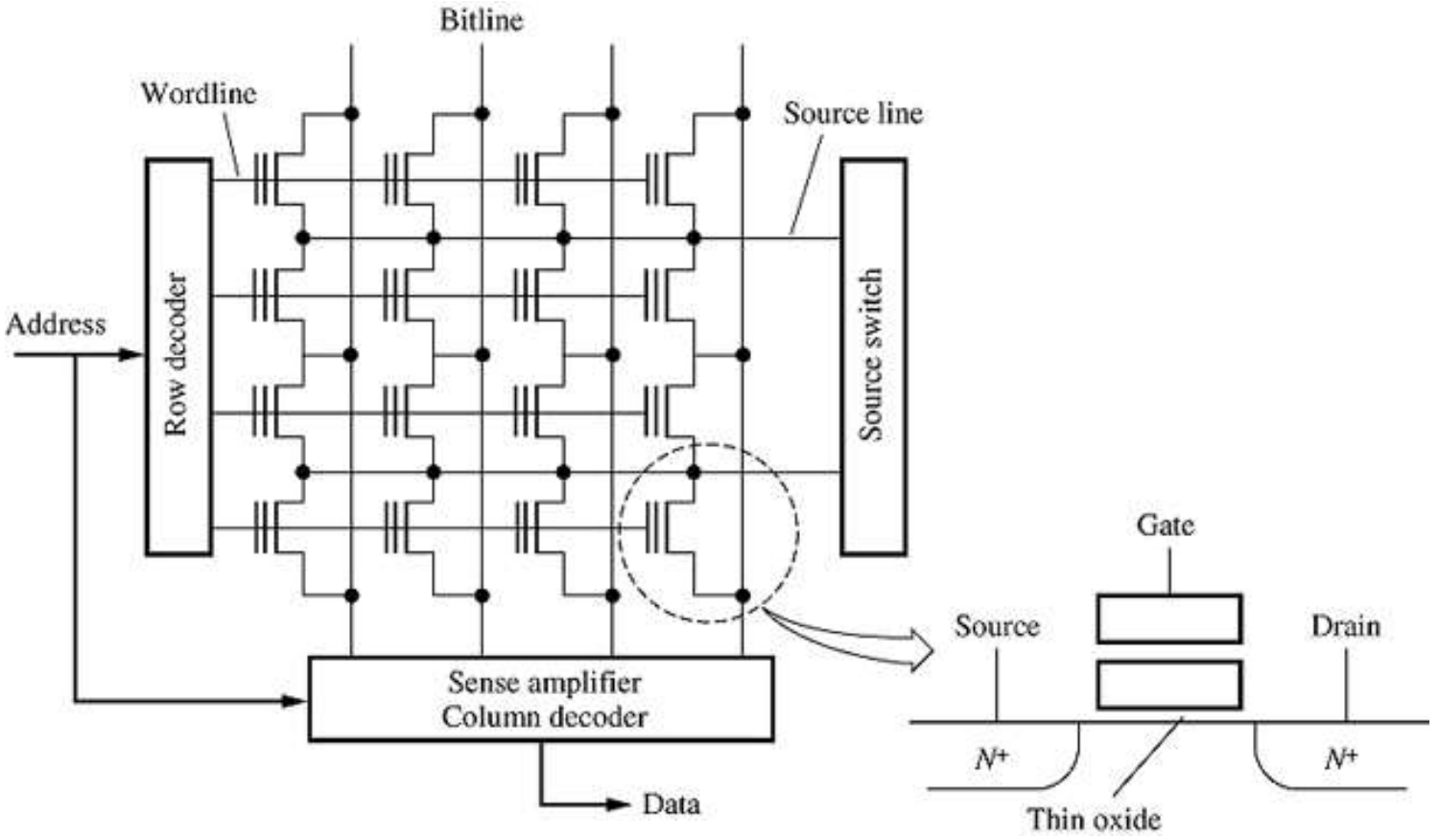
- Based on MOS transistor
- Transistor gate is redesigned
- Charge is placed on or removed from a 'floating gate'
- The threshold voltage V_{th} of the transistor is shifted by the (permanent) presence of this charge

Flash Memory

- NOR Flash
  - Combine the density of EPROM and versatility of EEPROM
  - NOR structure use HCI to program and use FN tunneling to erase
  - Erasure and programming times are slow due to need for precise control of the threshold
  - *Erasure performs in bulk* for a complete chip or sub-section of memory and takes between 100ms to 1 s
  - Perform V_T checking during erasure and adjust erasure time dynamically
  - Fast random read access time thus suitable for program-code storage

NOR flash

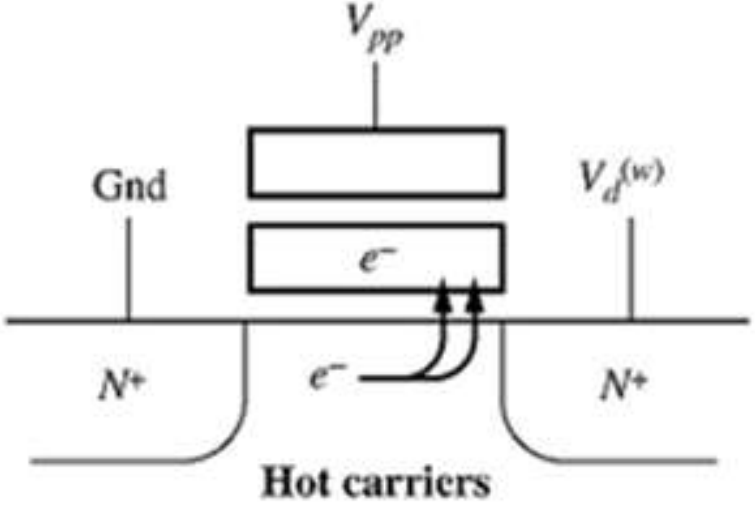
- NOR Flash architecture

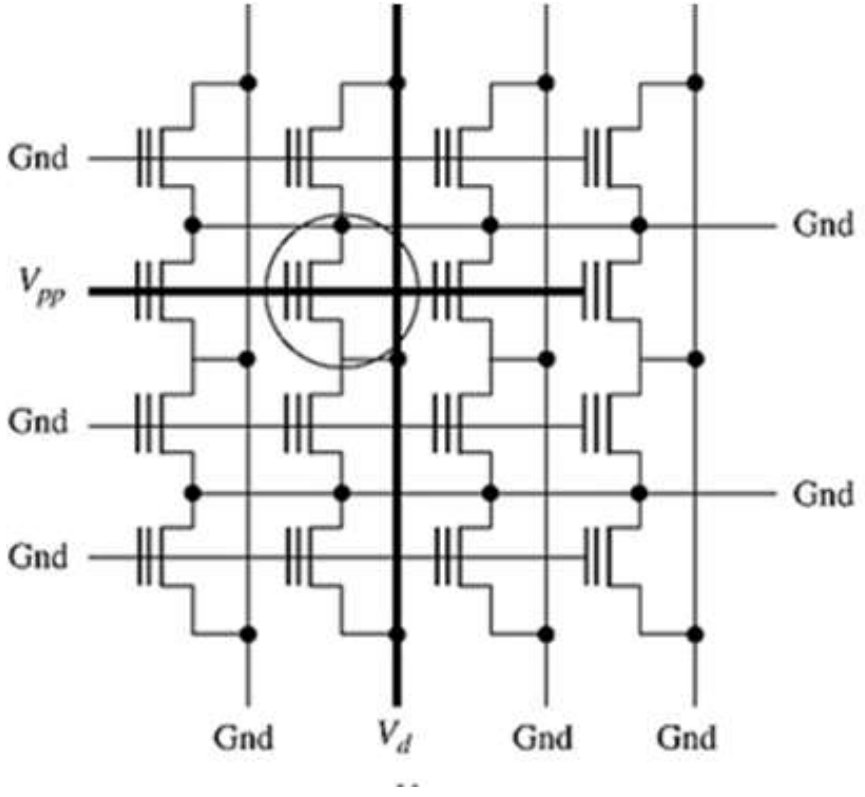


Source : Hodges

NOR Flash operation 1/3

- NOR Flash –Write
  - Use HCI to program: source voltage is connected to GND, then Vd is applied to selected bitline while Vpp is applied to selected wordline

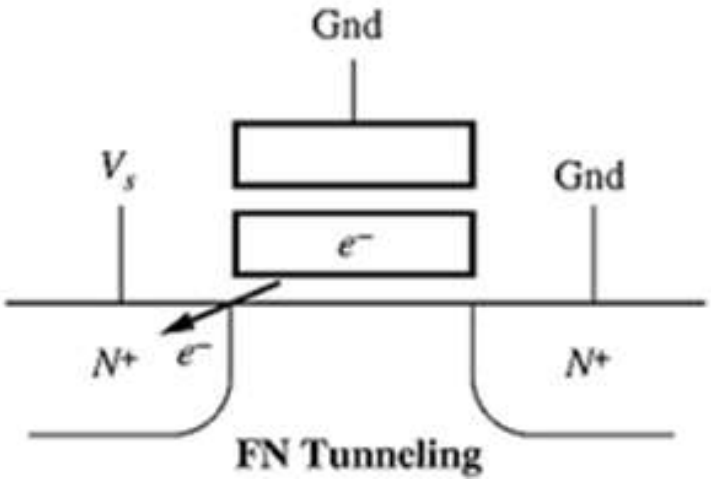




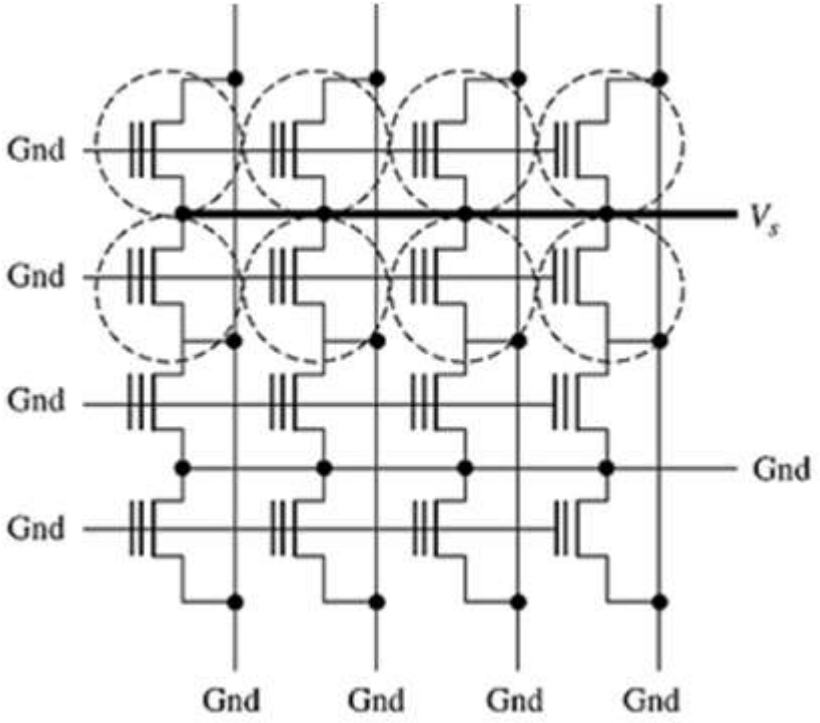
Source : Hodges

NOR Flash operation 2/3

- NOR Flash –Erase
  - Use FN tunneling: Apply GND to *all* Gate (WL), high voltage of Vs to the source node
  - Transistors connected to source connections are *all* erased at the same time

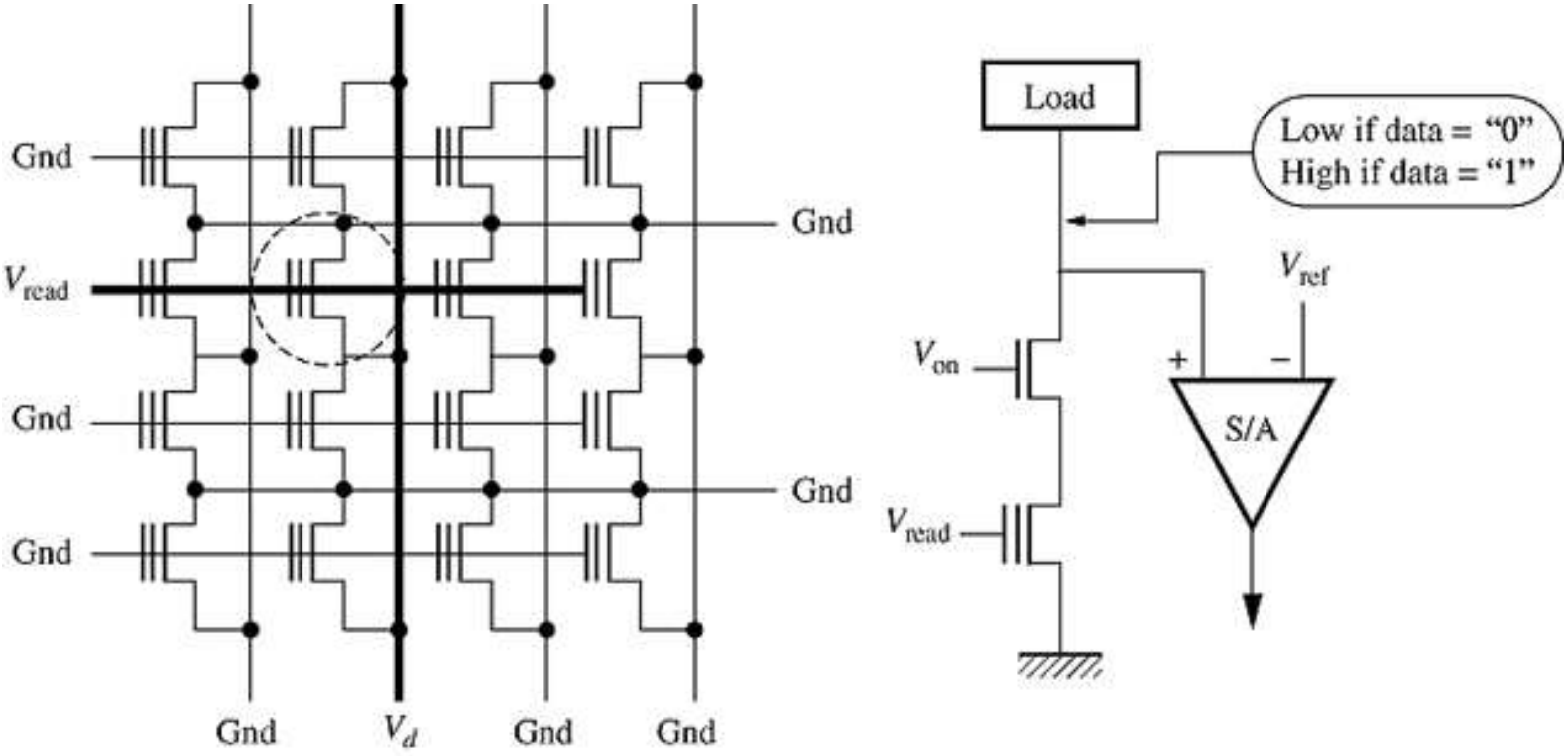


Source : Hodges



Nor Flash 3/3

- NOR Flash –Read
  - Apply GND to source connection, precharging the bitline to Vd and enable wordline with a Vread



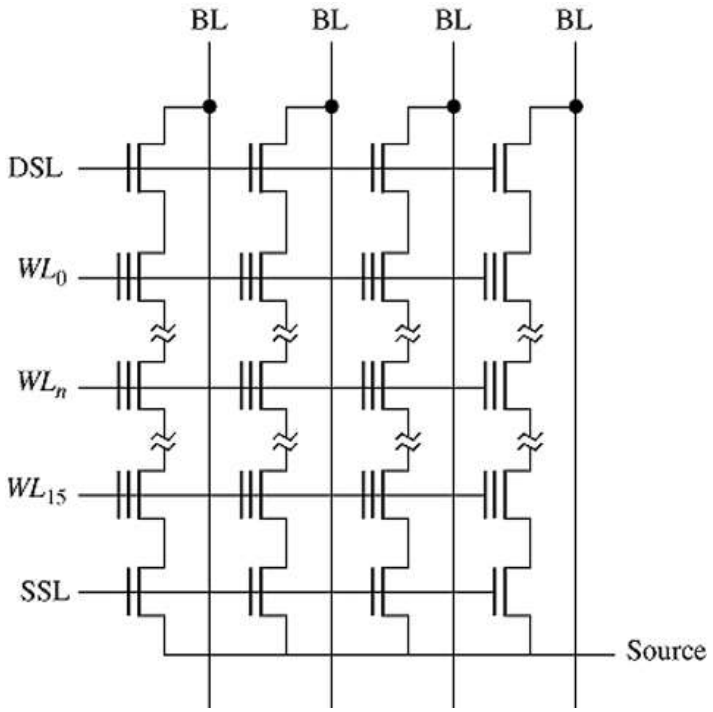
Source : Hodges

NAND Flash

- NAND Flash
  - 40% smaller and more dense than NOR array
  - Typically use FN tunneling for both write and erase which allows a much larger cycle limit usually more than 10^6 cycles
  - Fast write/erase and fast serial access but slower random access than NOR
  - Read operation is similar to NAND ROM
  - Suitable for applications that do not need fast random access such as video/audio file storage

NAND Flash Operation

- NAND Flash
  - Wordlines are normally high but one will go low and active low when decoder is activated
  - Erase: BL and Source are high, WL is GND results in negative V_T
  - Write “1”: SSL isolates source, then BL is GND and WL is high causing V_T to increase
  - Write “0”: Keeping BL high so V_T does not change
  - Read: SSL and DSL are enable. Then operate the same way as NAND ROM



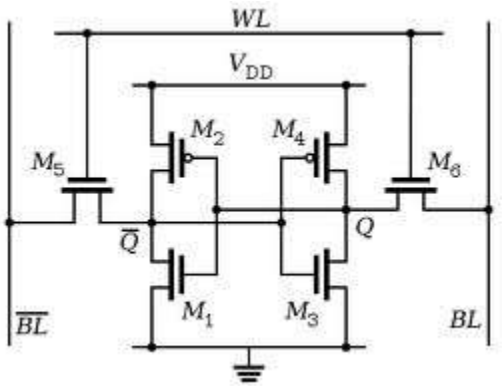
Source : Hodges

Standardization

- Part of the reason for the success of Flash memory
- Open NAND Flash Interface Working Group developed standard low-level interface
  - Standard pinout
  - Standard command set for reading, writing, and erasing NAND flash chips
  - Mechanism for self-identification
- Embedded FLASH – on the same chip as logic (does not need to be standardized)

Static RAM

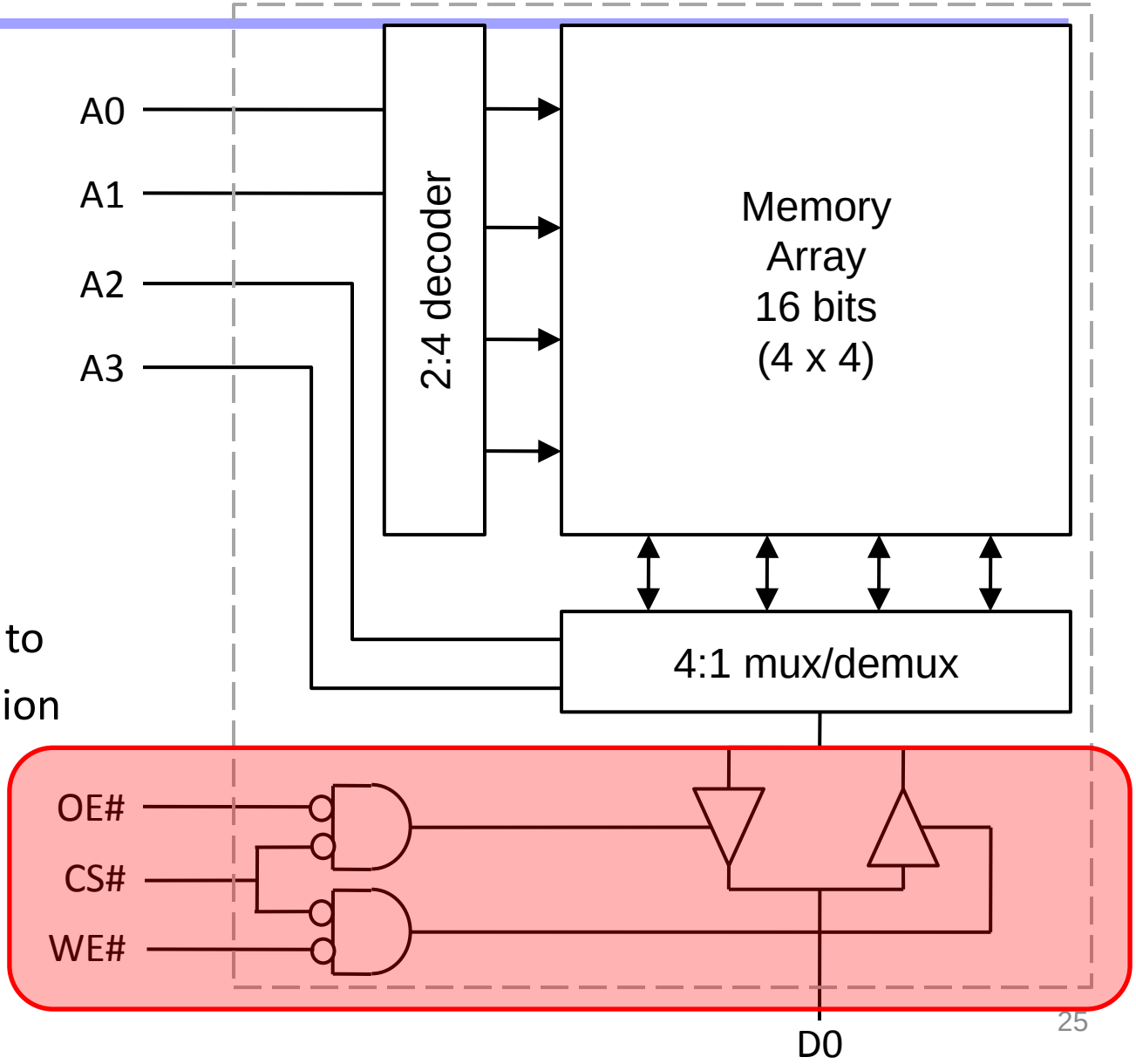
- SRAMs are volatile
- Basic cell
  - Bistable core
    - 4T: uses pullup resistors for M2, M4
    - 6T: uses P-FET for M2, M4
  - Access transistors
  - BL, BL# are provided to improve noise margin
- 6T is typically used (but has poor density)
- Fast access times O(10 ns)
- Read/write speeds are symmetric
- Read/write granularity is word



Memory-bus interface

Control signals

- Select chip
- Select memory cell
- Control read/write
- Map internal array to external configuration (4x4 → 16x1)



Memory-bus interface

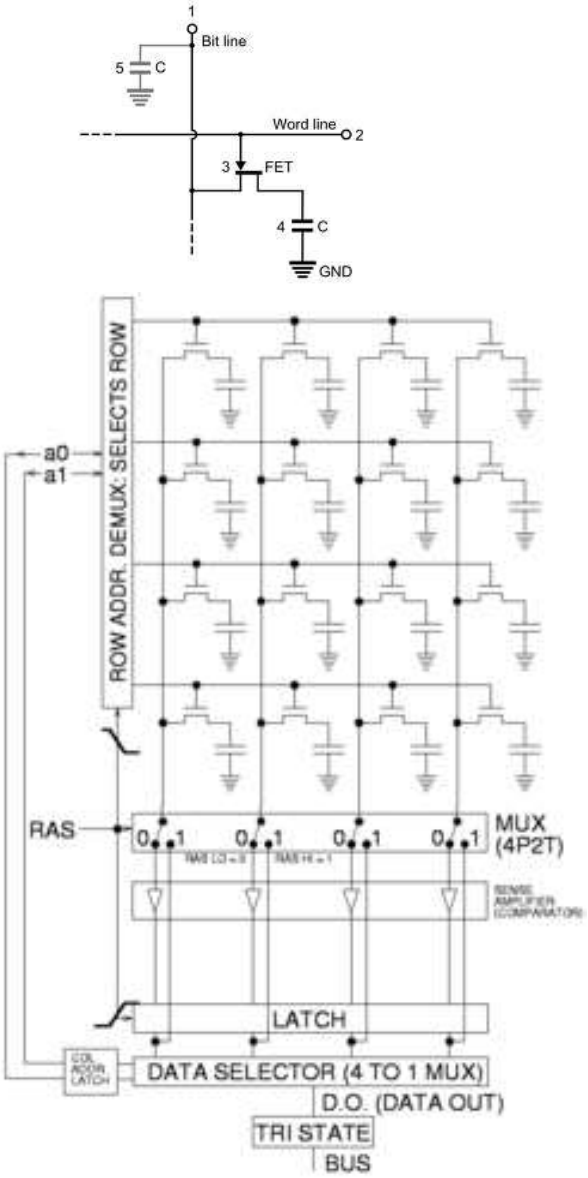
- Chip Select (CS#)
  - Enables device
  - Ignores all other inputs if CS# is not asserted
- Write Enable (WE#)
  - Enables write tri-state buffer
  - Store D0 at specified address
- Output Enable (OE#)
  - Enable read tri-state buffer
  - Drive D0 with value at specified address
- Embedded memories have a more complex bus interface

Dynamic RAM

- Requires only 1T and 1C per cell
- Outstanding density and low cost
- Compare to the 6T's per SRAM cell
- Cost advantage to DRAM technology

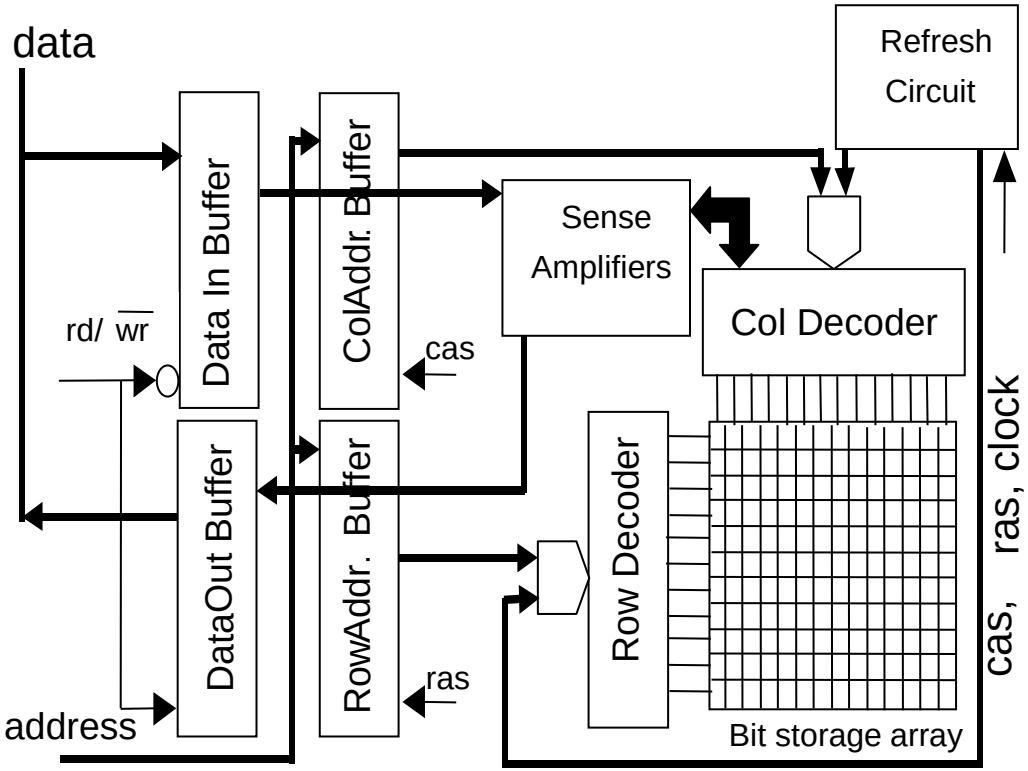
- Small charges involved → relatively slow
  - Bit lines must be pre-charged to detect bits
  - Reads are destructive; internal writebacks needed

- Values must be refreshed periodically
  - Prevents charge from leaking away
  - Complicates control circuitry slightly



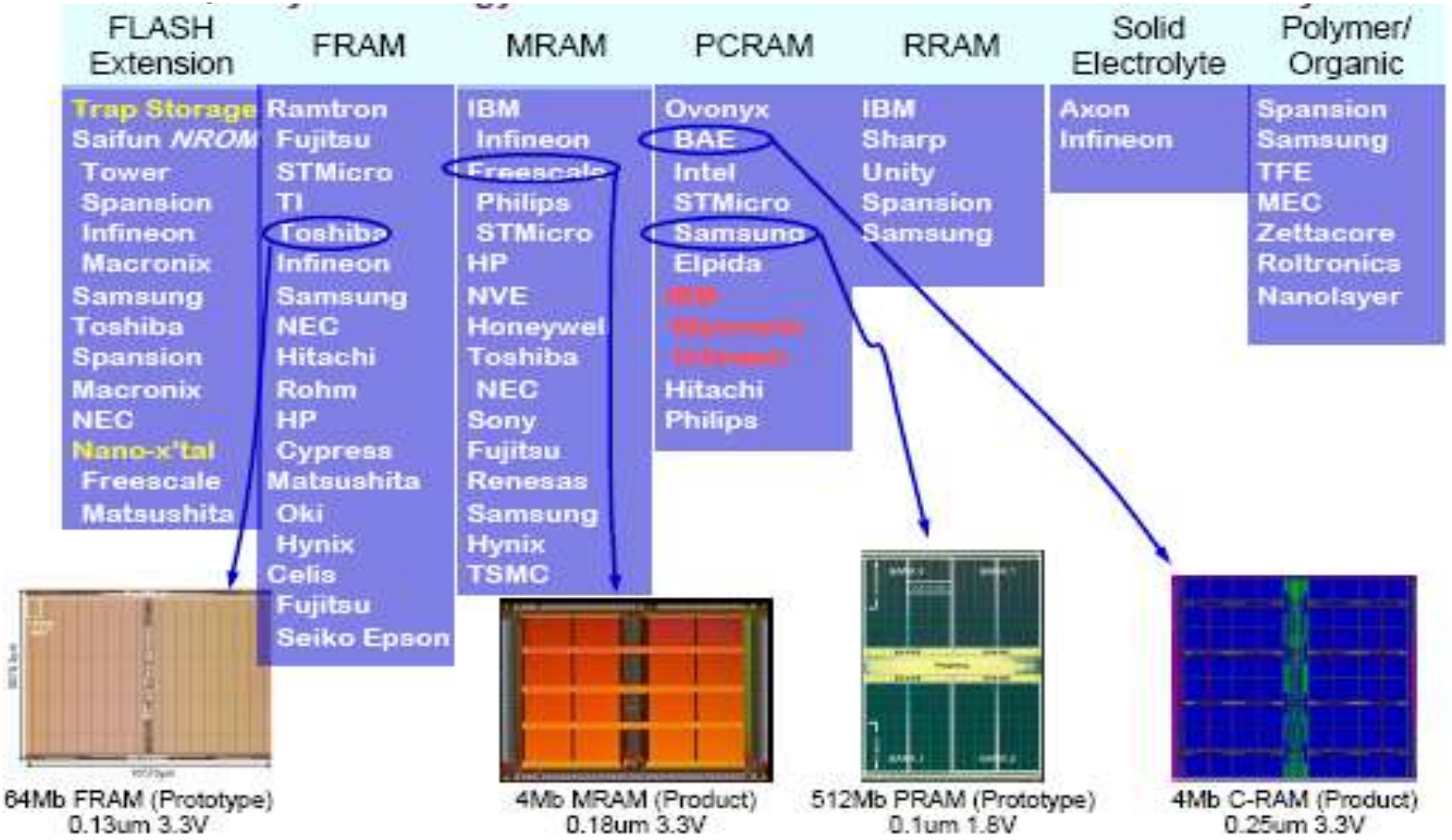
Basic DRAM architecture

- Address bus multiplexed between row and column components
- Row and column addresses are latched in, sequentially, by strobing ras and cas signals, respectively
- Refresh circuitry can be external or internal to DRAM device
  - strobes consecutive memory address periodically causing memory content to be refreshed
  - Refresh circuitry disabled during read or write operation



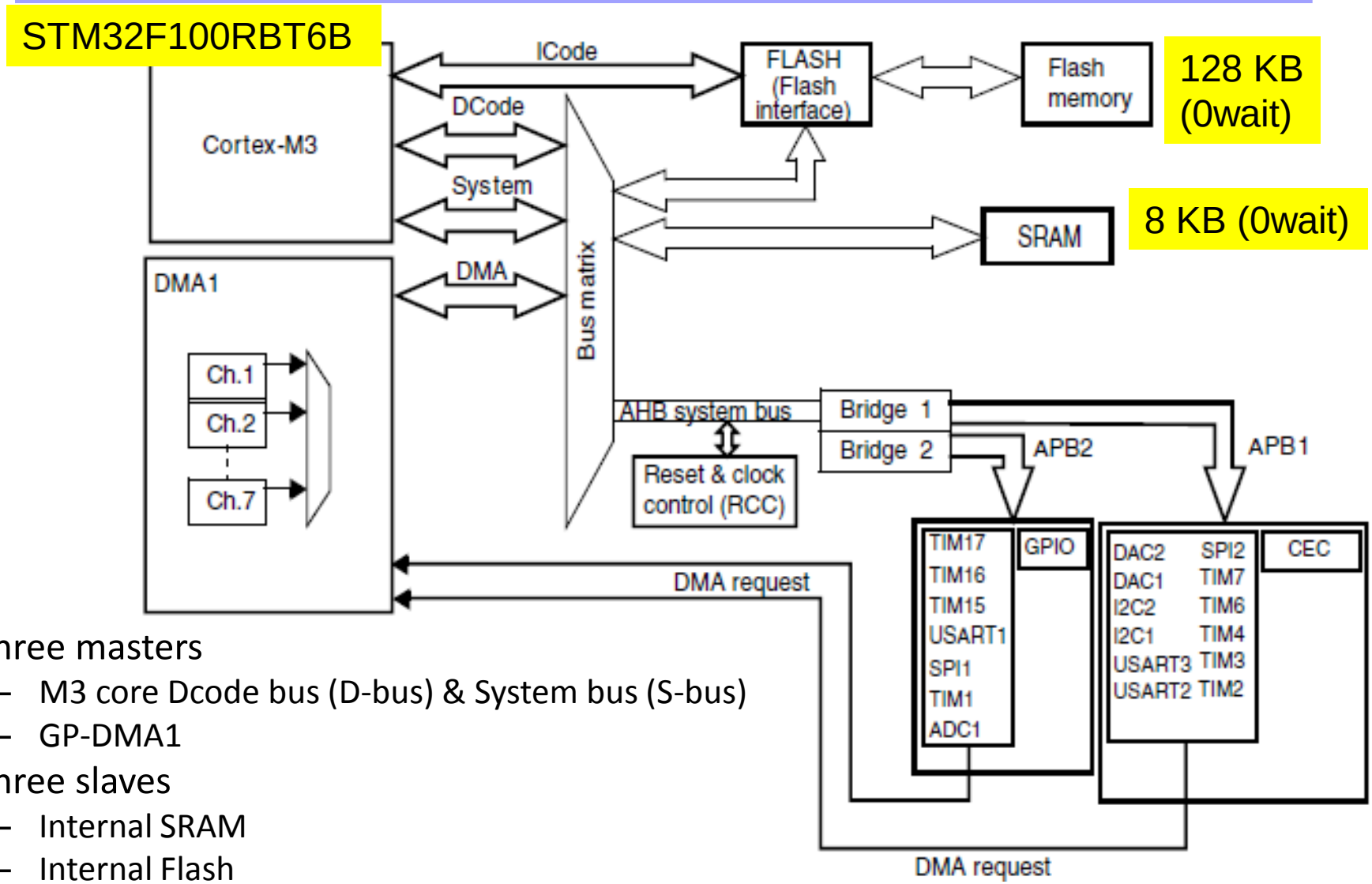
Emerging Memory Technologies

- Memory technology is an active area for the



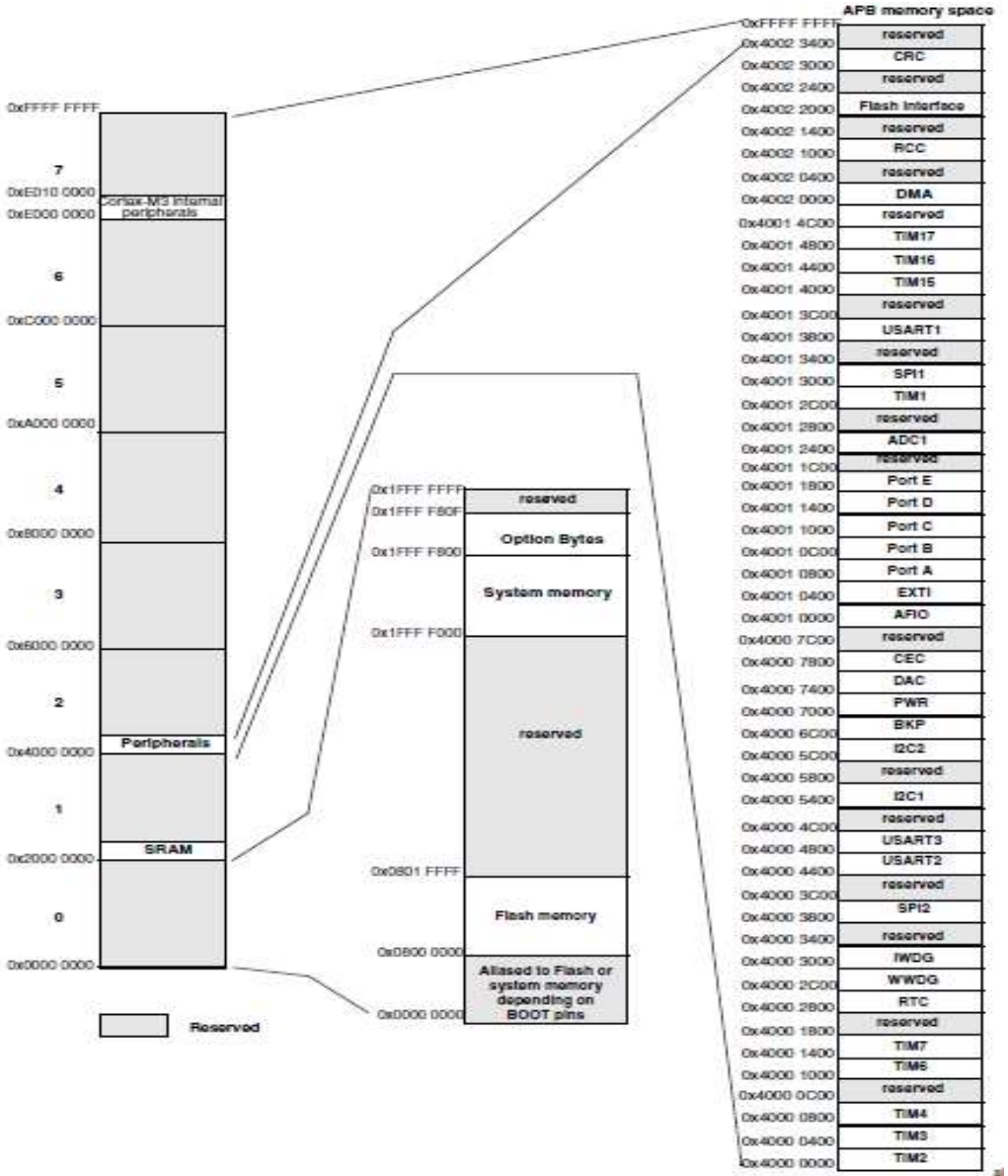
Embedded Memory for STM32F100RBT6B

STM30F100xxx System Architecture



- Three masters
  - M3 core Dcode bus (D-bus) & System bus (S-bus)
  - GP-DMA1
- Three slaves
  - Internal SRAM
  - Internal Flash
  - AHP to APB bridges to APB peripherals

STM32F100RBT6B Memory map



Programming the Flash memory

- The in-circuit programming (ICP) method is used to update the entire contents of the Flash memory, using the JTAG, SWD protocol or the boot loader to load the user application into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.
- In-application programming (IAP) can use any communication interface supported by the microcontroller (I/Os, USB, CAN, UART, I2C, SPI, etc.) to download programming data into memory. IAP allows the user to re-program the Flash memory while the application is running. Nevertheless, part of the application has to have been previously programmed in the Flash memory using ICP.

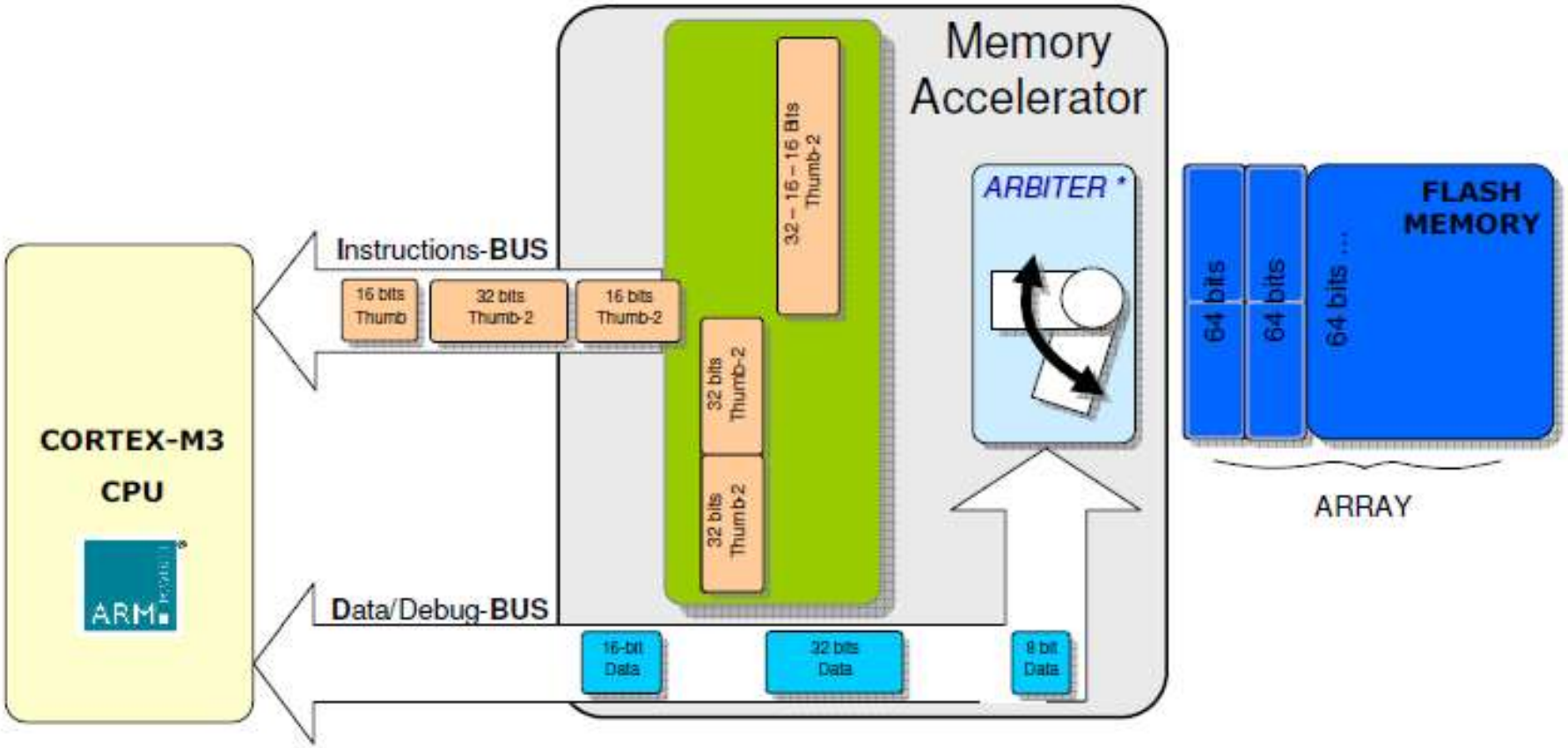
FLASH Features

- Flash Features:
  - Up to 128KBytes
  - 1 KByte Page size
  - Endurance: 10k cycles
  - Memory organization:
    - Main memory block
    - Information block
  - Access time: 35ns
  - Halfword (16-bit) program time: 52.5 µs (Typ)
  - Page / Mass Erase Time: 20ms
- Flash interface (FLITF) Features:
  - Read Interface with pre-fetch buffer
  - Option Bytes loader
  - Flash program/Erase operations
  - Types of Protection:
    - Readout Protection
    - Write Protection

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	. . .	. . .	. . .
	Page 127	0x0801 FC00 - 0x0801 FFFF	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

Flash Mem Accelerator

- Supports 72MHz operation from Flash
- 64-bit wide Flash with prefetch (2x64bits buffers)



Information Block

- The Information Block consists of:
  - 2 KBytes for SystemMemory : contains embedded Bootloader
  - 16 Bytes for Small Information block (SIF): contains The Option bytes
- 8 option bytes (SIF Block) are available :
  - 4 for write protection
  - 1 for read protection
  - 1 for Device configuration:
  - IWDG HW/SW mode
  - Reset when entering STANDBY mode
  - Reset when entering STOP mode
  - 2 For User Data (To store Security IDs, etc.)
- After unlocking the FPEC, the user has to authorize the small info block programming by writing 2 key values then he can program the Option bytes
- On every reset, the option bytes loader performs a read of the information block and stores the data into the FPEC registers (when programmed the option bytes are taken into account only after reset)

System Memory

- System memory is used to boot the device in System memory boot mode.
- The area is reserved for use by STMicroelectronics and contains the boot loader which is used to reprogram the Flash memory using the USART1 serial interface. It is programmed by ST when the device is manufactured, and protected against spurious write/erase operations

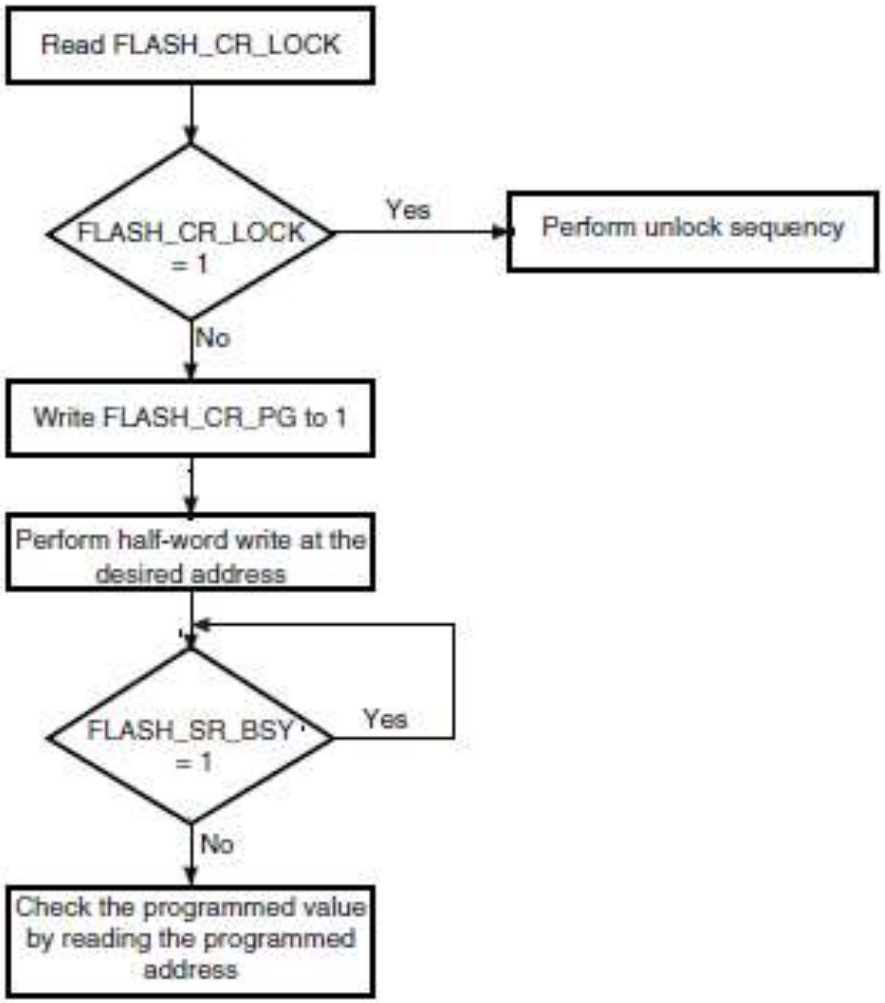
Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	. . .	. . .	. . .
	Page 127	0x0801 FC00 - 0x0801 FFFF	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

FLASH Control Interface

- The Flash program and erase operations are handled by the Flash program and erase controller (FPEC)
  - After reset the FPEC is protected, an unlocking sequence should be performed (write of 2 key values) to unlock the Flash
  - The Flash can be programmed with 16-bits at a time
  - Flash can be erased page-wise or completely: Mass Erase
  - I-bus stalled during program\erase
- The Read access can be performed with the following configuration:
  - Latency: Number of wait state for a read operation programmable on the fly
  - Prefetch buffer of 2x64bit: For faster CPU execution can be enabled and disabled on the fly
  - Half Cycle: Flash access can be made on a half cycle of the HCLK to reduce power consumption, enabled by software

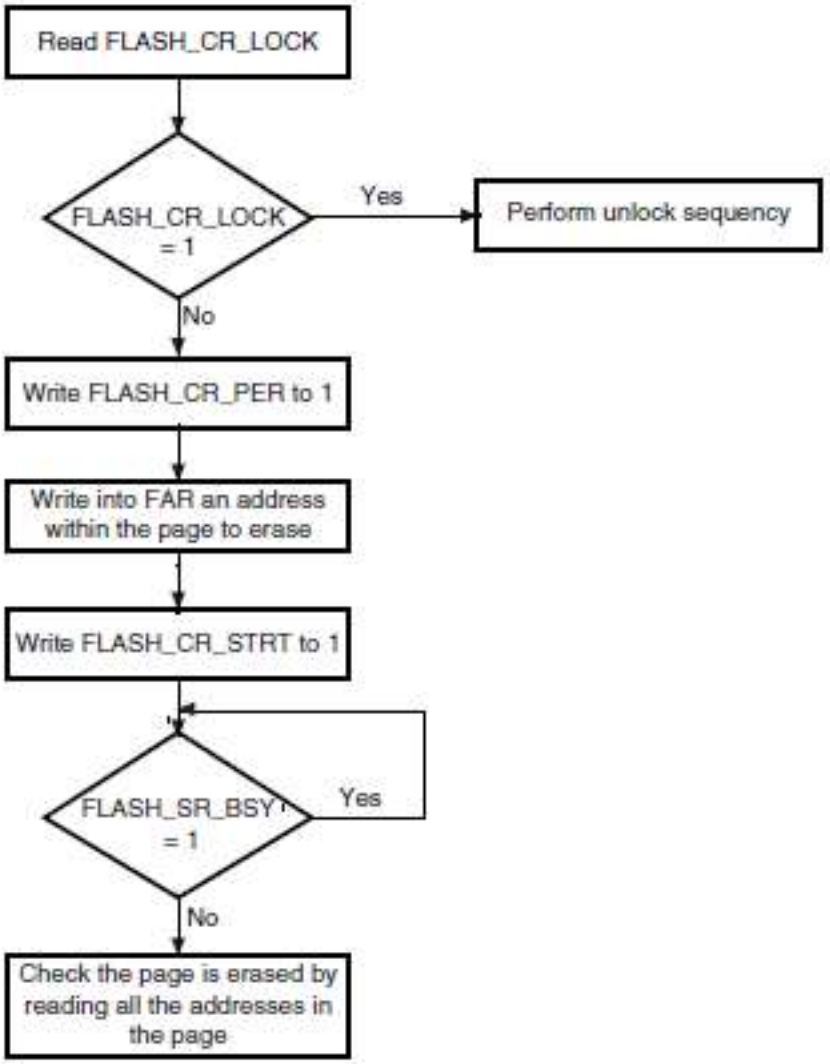
FLASH Write

- Note: Processor is not locked while write takes place



Page Erase

- Note: also mass erase is possible



Flash protection 1/2

- Two kind of protections are available:
  - Write protection to avoid unwanted writings
  - Readout protection to avoid piracy
  - Activated by setting option bytes in the Small Information Block (SIF)
- Write Protection
  - The write protection is implemented with a choice of protecting 4 pages (4K) at a time
  - A total of 4 Options bytes are used to protect all the 128K main Flash
  - Any programming or erase of a protected page is discarded and the Flash will return protection error flag on FSR status register
  - Unprotection:
    - Erase the entire Small Information block
    - The result of Readout protection code (RDP) will be 0xFF, the readout protection is enabled
    - Program the correct code 0xA5 of RDP to disable read protection
    - Reset the device (System Reset) to re-load the options bytes for disabling any write protection

Flash Protection 2/2

- Readout protection
  - When This protection is enabled :
    - Main Flash memory read access is not allowed except for the user code (when booting from main Flash memory itself with the debug mode not active).
    - Pages 0-3, are automatically write-protected. The rest of the memory can be programmed by the code executed from the main Flash memory (for IAP, constant storage, etc.), but it is protected against write/erase (but not against mass erase) in debug mode or when booting from the embedded SRAM.
    - All features linked to loading code into and executing code from the embedded SRAM are still active (JTAG/SWD and boot from embedded SRAM) and this can be used to disable the read protection.
- Unprotection:
  - Erase the entire Small Information block
  - The result of Readout protection code (RDP) will be 0xFF, the read protection will be still enabled
  - Program the correct code 0xA5 of RDP to unprotect the memory, this operation will first force a Mass Erase of the main block
  - Reset the device (POR Reset) to re-load the options bytes (and the new RDP code), and disable the Readout protection