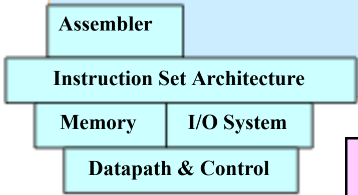
ELEC 3300
Introduction to Embedded Systems

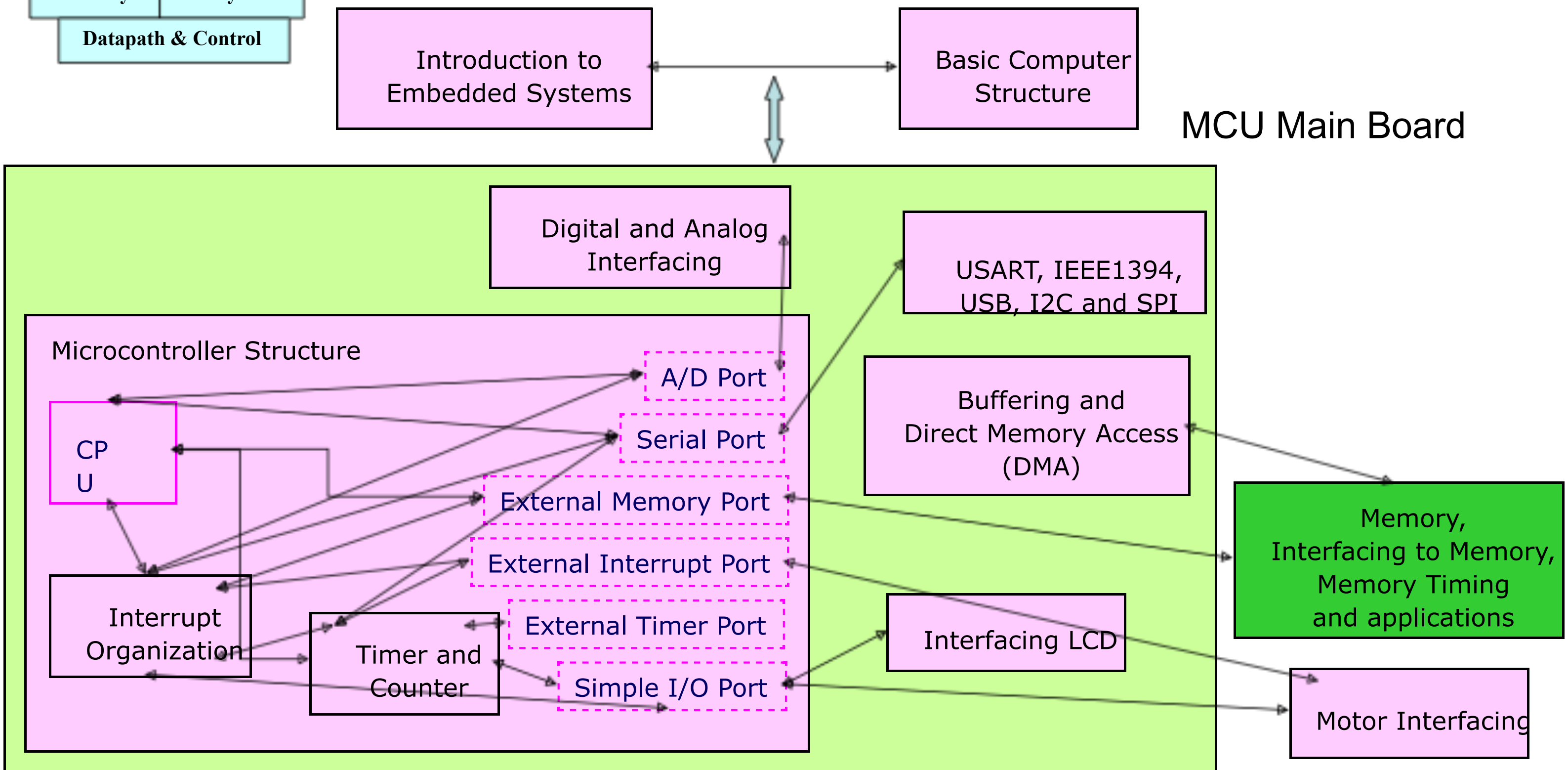
Topic 10

Memory, Interfacing to Memory, Memory Timing, and Applications

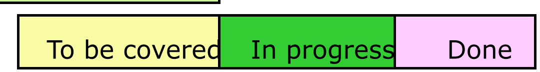
Prof. Vinod Prasad

Course Overview





In this course, STM32 is used as a driving vehicle for delivering the concepts.



Expected Outcomes

- On successful completion of this topic, you will be able to
  - Summarize the types of memory technologies
  - Interpret both hardware and software interfacing including
    - CPU memory interface
    - Timing diagrams of memory accesses (Read / Write / Refresh Operations)
  - Explain key considerations of memory architectures including
    - Memory Expansion
    - Paging

Memory Technologies

FLASH

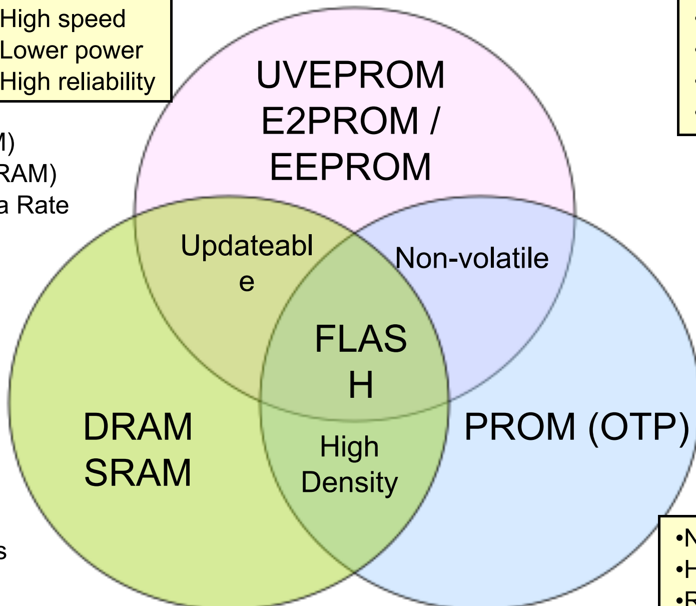
- High density
- Low cost
- High speed
- Lower power
- High reliability

DRAM (Dynamic RAM)
SDRAM (Synchronous DRAM)
DDR SDRAM (Double Data Rate SDRAM)

- High density
- Low cost
- High speed
- Lower power
- Refreshing is required

SRAM (Static RAM)
SSRAM (Synchronous SRAM)

- Low density
- Low cost
- High speed
- Lower power



UVEPROM
(UV Erasable Programmable ROM)

- Non-volatile
- High density
- Erasable by UV light (takes an hour)
- programmed many times (~100)

EEPROM
(Electrically Erasable Programmable ROM)

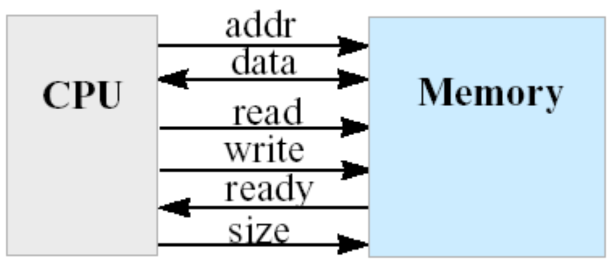
- Non-volatile
- Lower density
- Lower reliability
- Higher cost
- Erasable by electrical (takes few ms)
- programmed many times (~1000)

PROM (Programmable ROM)

- Non-volatile
- High density
- Reliable
- Low cost
- One-time programmable (OTP)
(suitable for mass production with stable code)

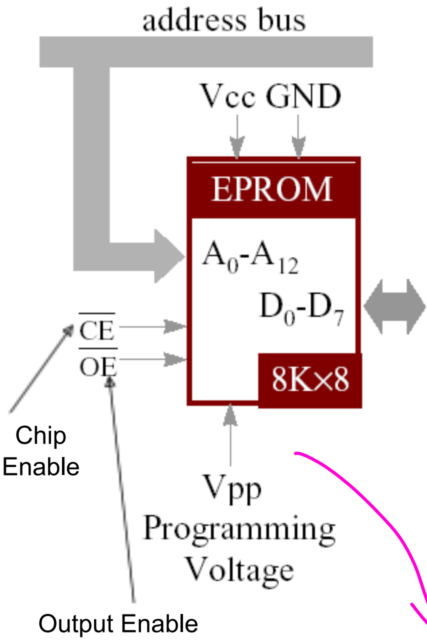
CPU-Memory Interface

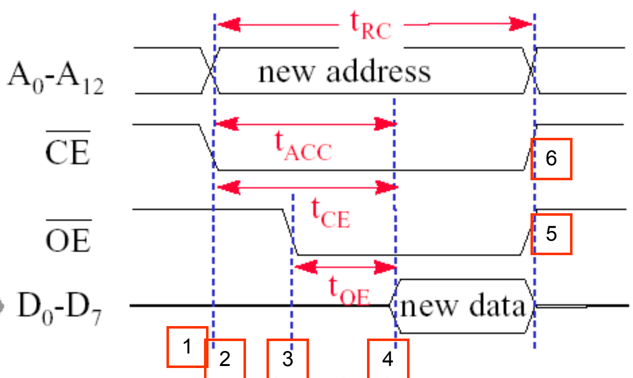
- CPU-Memory interface usually consists of
  - Unidirectional address bus
  - Bidirectional data bus
  - Read/write control lines
  - Ready control line
  - Size (byte, word) control line



- Memory access involves memory bus transaction
  - Read: set address, read and size, copy data when ready is set by memory
  - Write: set address, write and size, done when ready is set

Example 1: EEPROM Timing for Read Access





t_{RC} = read cycle time

t_{ACC} = address access time

t_{CE} = chip select access time

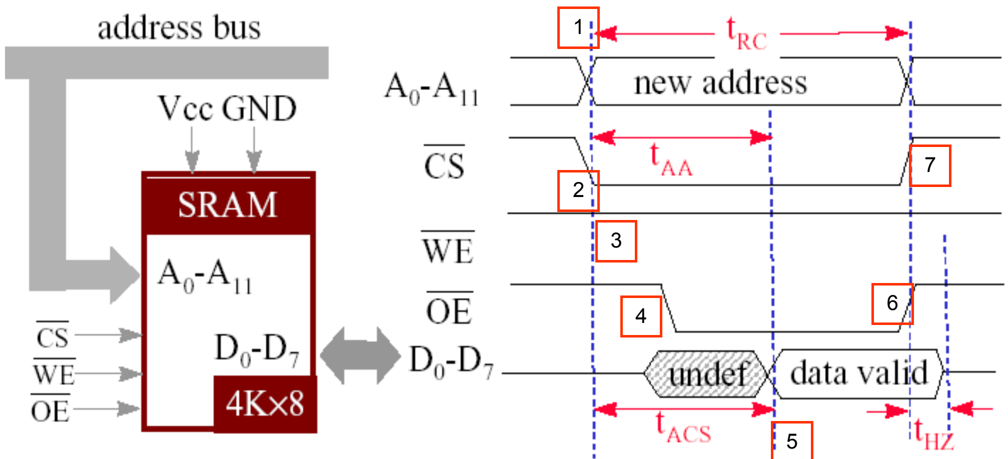
t_{OE} = output enable access time

		Pins			
		$\overline{CE}$	$\overline{OE}$	V_{pp}	D_0-D_7
Mode	Program	0	1	12.5V	D_{in}
	Read	0	0	5V	D_{out}
	Standby	1	X	5V	Off

Step 1: send the address to EEPROM
Step 2: chip enable ($\overline{CE}$ = 0)
Step 3: output enable ($\overline{OE}$ = 0)
Step 4: data is ready
Step 5: output disable ($\overline{OE}$ = 1)
Step 6: chip disable ($\overline{CE}$ = 1)

chip enable by low signa

Example 2a: SRAM Timing for Read Access



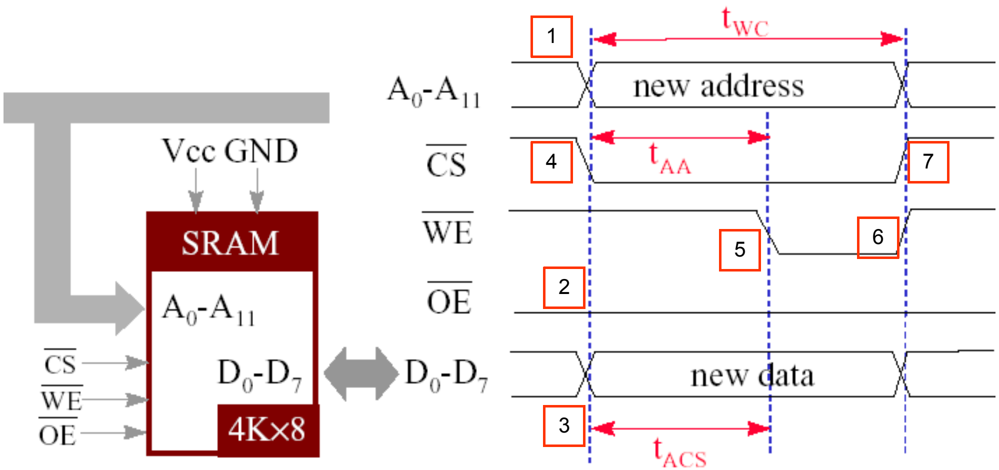
Address and Chip Select signals are provided t_{AA} time before data is available.

t_{RC} = read cycle time
t_{AA} = address access time
t_{ACS} = chip select access time
t_{HZ} = chip deselections to floating out

Step 1: send the address to SRAM
Step 2: chip enable ($\overline{CS}$ = 0)
Step 3: read enable ($\overline{WE}$ = 1)
Step 4: output enable ($\overline{OE}$ = 0)
Step 5: data is ready
Step 6: output disable ($\overline{OE}$ = 1)
Step 7: chip disable (CE = 1)

$\overline{WE}$: Write/Read Enable (0 for Write and 1 for Read)

Example 2b: SRAM Timing for Write Access



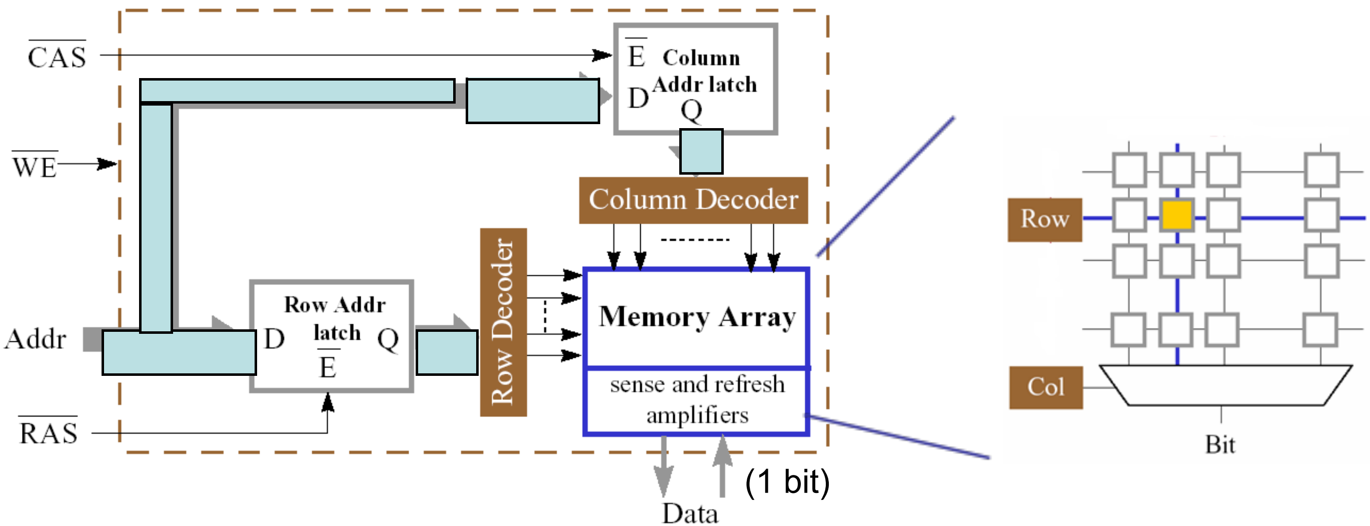
Address and Data must be stable t_{AA} time before write enable signal falls.

t_{WC} = write cycle time
t_{AA} = address access time
t_{ACS} = chip select access time

Step 1: send the address to SRAM
Step 2: output disable ($\overline{OE}$ = 1)
Step 3: data ready (D0 – D7)
Step 4: chip select ($\overline{CS}$ = 0)
Step 5: write enable ($\overline{WE}$ = 0)
Step 6: write disable ($\overline{WE}$ = 1)
Step 7: chip disable (CE = 1)

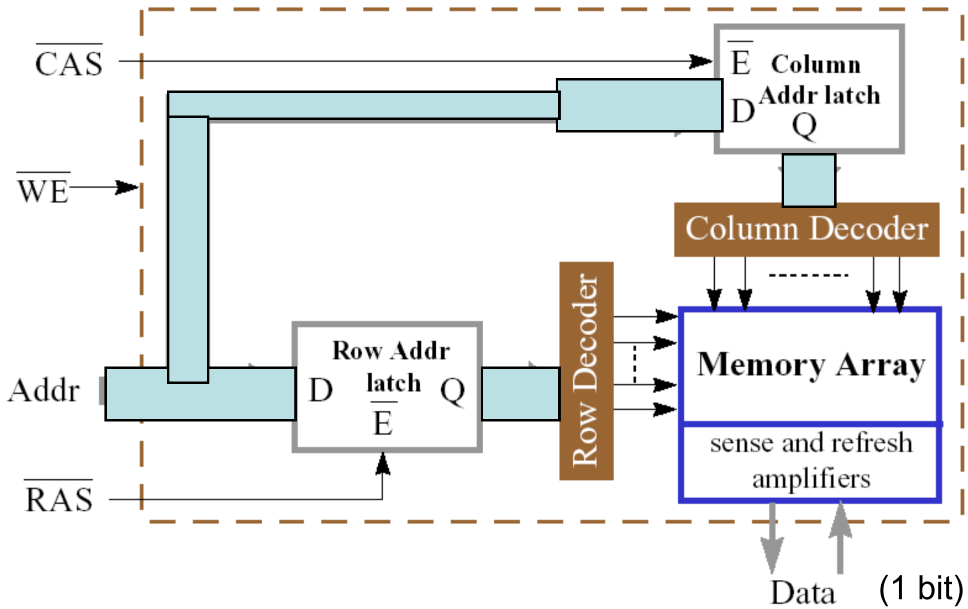
DRAM Memory Devices

- DRAM devices differ from SRAMs in these ways
  - The data is stored as charge on a capacitor (which leaks away unless refreshed for every 2ms)
  - The organization is usually ‘bit-wide’ instead of ‘byte-wide’
  - DRAMs are organized internally as a matrix of storage bits with each bit having a row address and column address and multiplexing of these addresses is used



Basic DRAM Operation

- Address is time multiplexed into separate row and column address latches
- Row and column addresses are decoded and strobed sequentially by RAS and CAS signals
- Each sense amplifier both reads and restores the data to the bit cell



(1 bit)

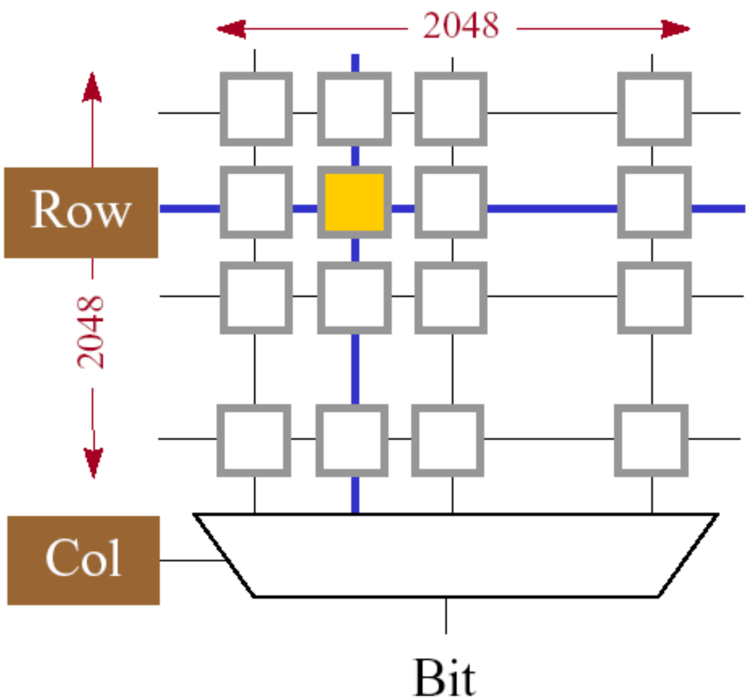
(RAS = 0, CAS = 1)

(RAS = 1, CAS = 0)

Sense amplifier senses the low power signals from a bitline that represents a data bit stored in a memory cell, and amplify the small voltage swing to recognizable logic levels so the data can be interpreted properly by logic outside the memory.

Basic DRAM Operation

Example: 4M x 1bit DRAM chip.



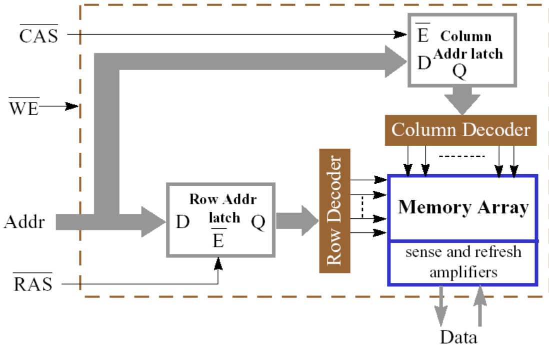
For this chip, the row and column addresses are 11 bits each. $2^{11x2} = 4Mx1$ bits.

The 11 least-significant addr bits (A10-A0) are set up on the addr pins and latched into the row address latch.

Then the 11 most-significant addr bits (A21-A11) are set up and latched into the column address latch.

The row & col address decoders selects the 1 bit out of 2048x 2048 and feeds it to the data output.

DRAM: Read Timing Diagram



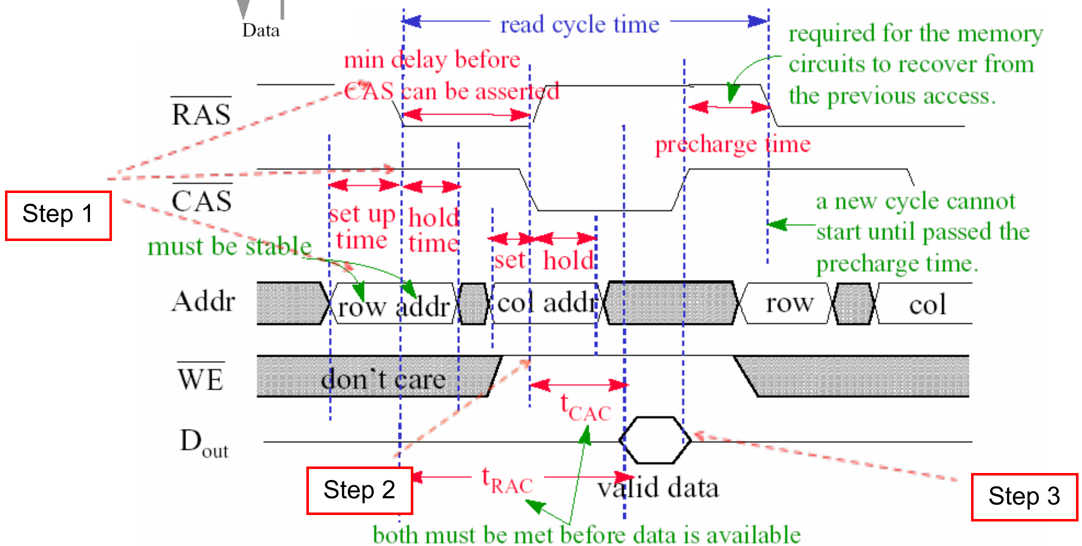
Step 1: Send the address

- row address □ row decoder (RAS = 0
- column address □ column decoder

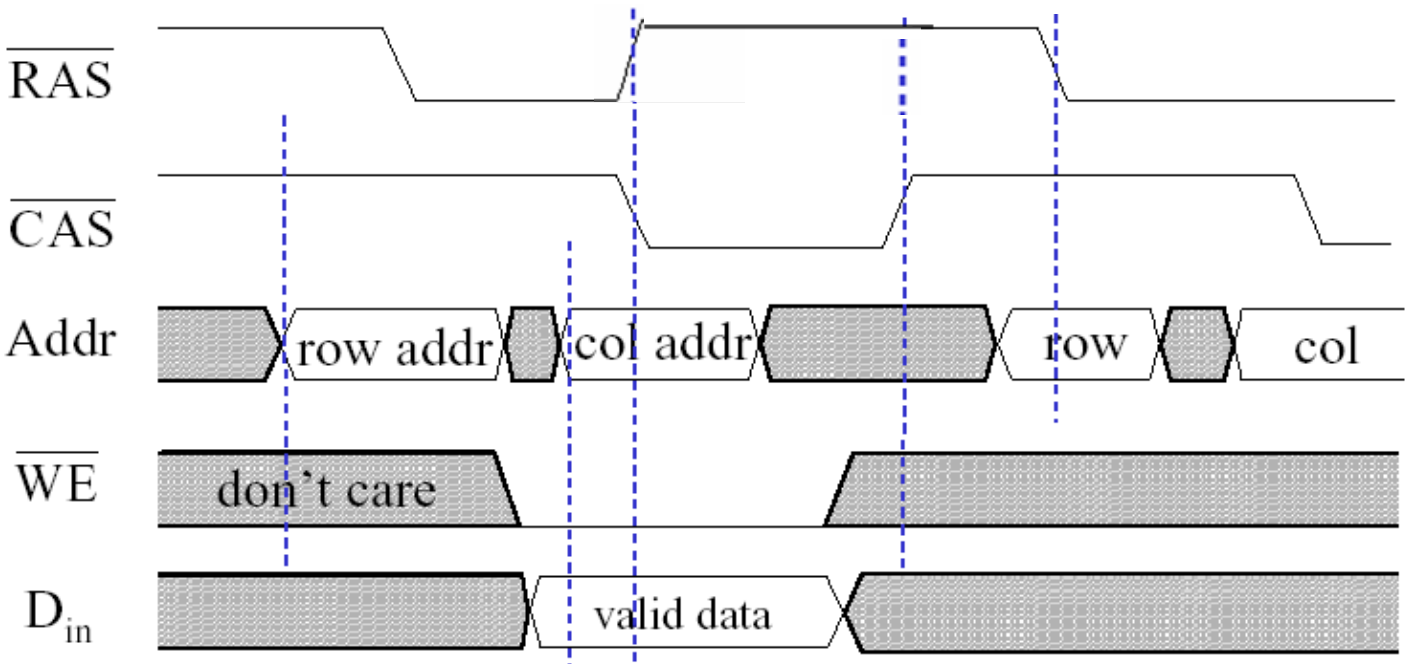
Step 2: Chip select (if necessary)

Step 2: Read enable (WE = 1)

Step 3: Data ready



DRAM: Write Timing Diagram



DRAM: Tasks

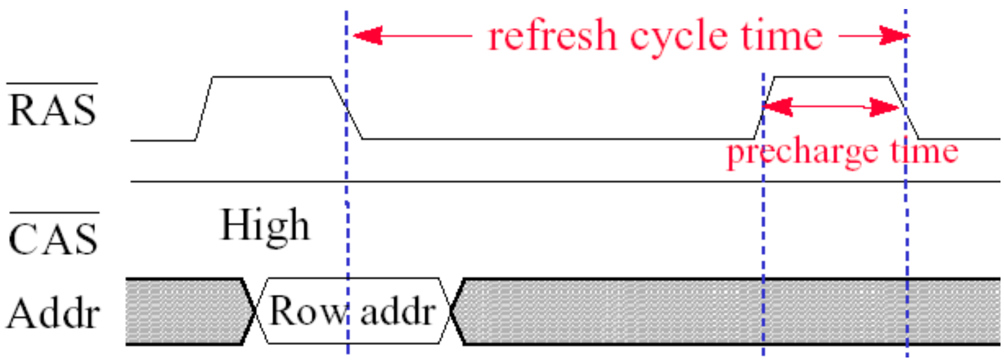
Activate: Opens a row of a bank. A row must be active for reading and writing data. If a row is idle, and if a row is activated it stays that way until a precharge command.

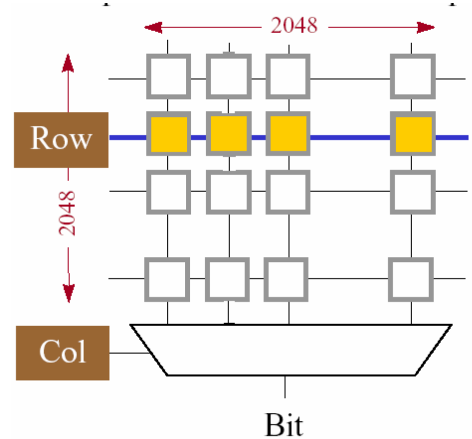
Precharge: Closes the open row, putting them into the idle state. Data is still stored in idle banks, but they must be activated again before reading or writing.

Read and Write: Data Read and Write.

Refresh: Refreshes the charge in memory cells by writing data back in place without changing it. DRAM is volatile memory, which means that it requires power to store data: bits are represented by charges on capacitors, which leak over time if they are not read or written to.

DRAM: Refresh Timing Diagram





- Refresh: read out the voltage of each cell, amplify it, charge the capacitor back to the original voltage
- All cells in a row are refreshed in parallel
- The entire DRAM is refreshed by providing each possible row address in sequence by a counter to increment the row address at each refresh cycle
- For a 4Mx1-bit DRAM, 2048 refresh cycles are needed

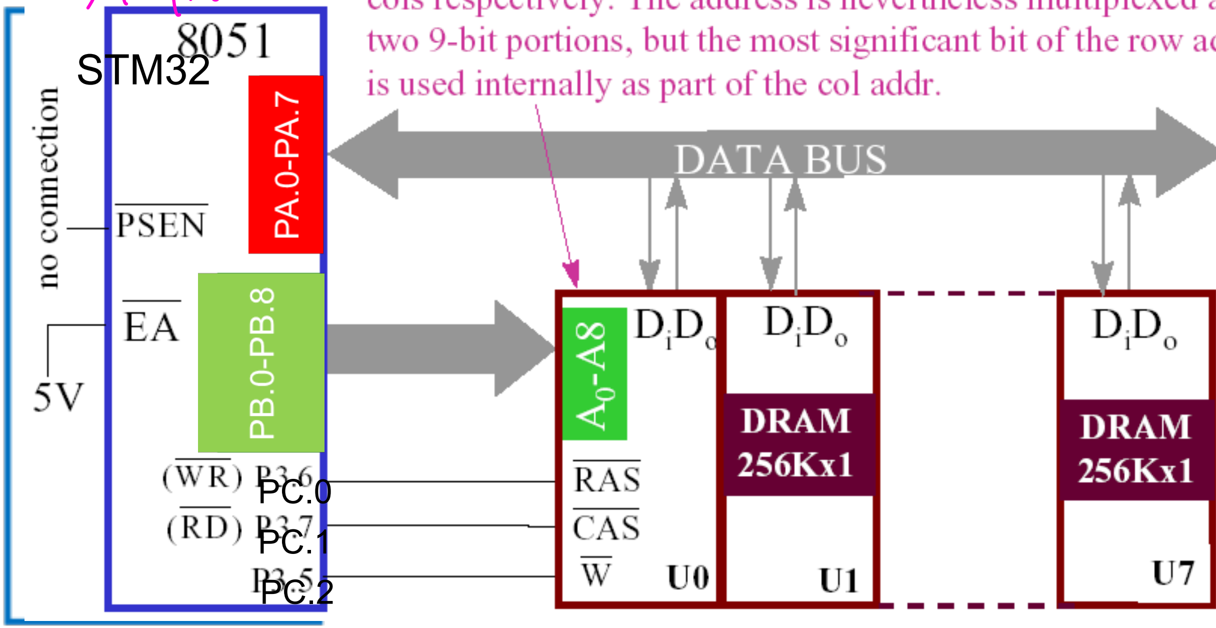
Precharge time: The minimum number of clock cycles required between issuing the precharge command and opening the next row.

STM32: Interfacing to DRAM

1 dram → bit access

8 dram → 1 bit x8 =

Note that this 256K DRAM is organised as 256 rows by 1024 cols. So, 8bits and 10bits are required to decode the rows and cols respectively. The address is nevertheless multiplexed as two 9-bit portions, but the most significant bit of the row addr is used internally as part of the col addr.



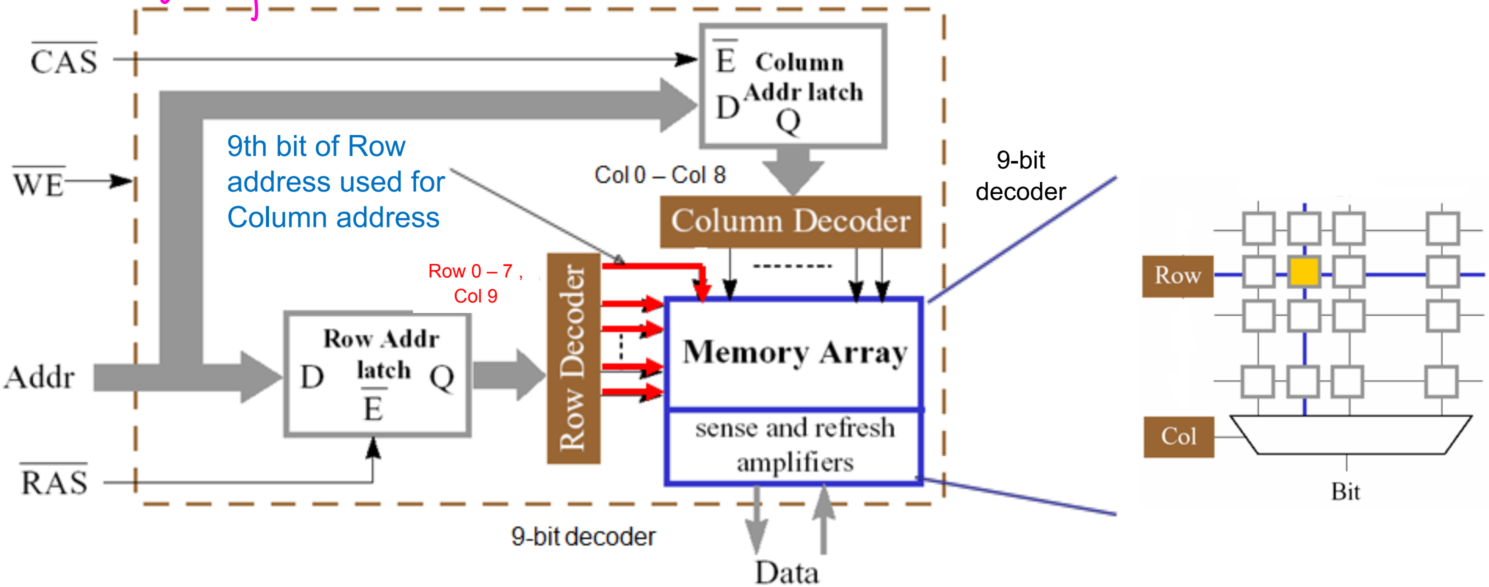
9-bit Row Address Decoder: A0 -A8: Row0 - Row7, Col9

9-bit Col Address Decoder: A0 -A8: Col0 - Col8

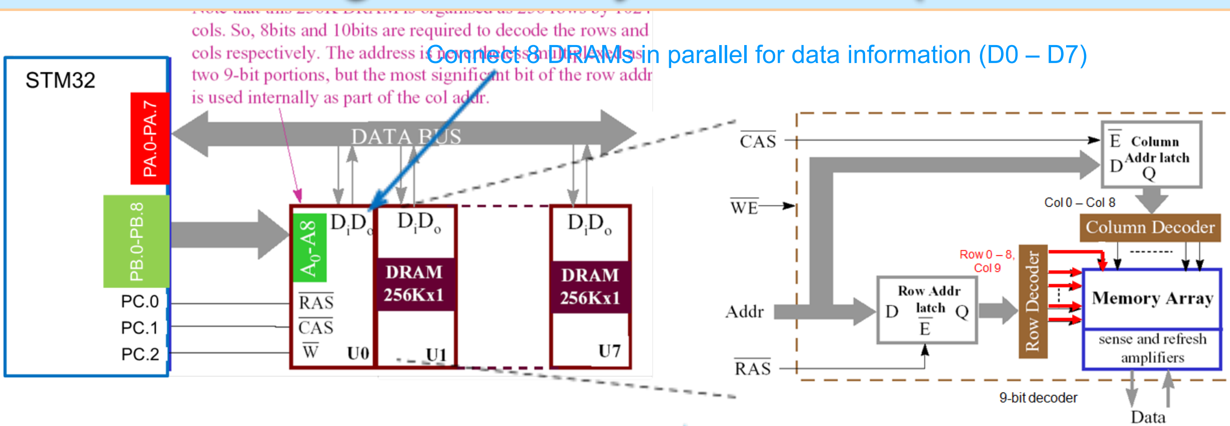
STM32: Interfacing to DRAM

- Internal architecture of 256K DRAM: 256 rows 1024 cols

noise
charge leak < 5V (1 bit) → which is why we need an amplifier.



STM32: Controlling DRAM by Software (Read Access)



Assume:

Address information:

ColAdd = 0x03AB; (10 bits)

RowAdd = 0x00F7; (8 bits)

Control information:

PC.2 = 1 ; /W

PC.0 = 1 ; /RAS high

PC.1 = 1 ; /CAS high

Data information

PA.0 (D0) – PA.7 (D7)

DRAM Memory Array

Row Address (8 bits) Row 7 , Row 6, ……, Row 0	Column Address (10 bits) Col 9, Col 8, ……, Col 1, Col 0

Hardware Decoder

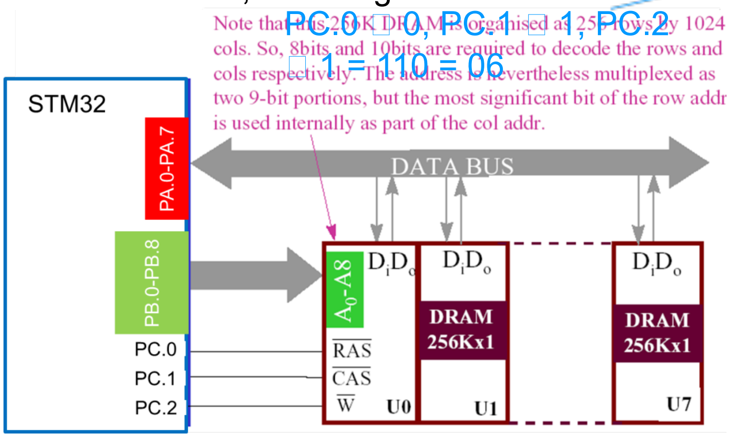
Row Decoder (9 bits) Row 0 – Row 7, Col 9	Column Decoder (9 bits) Col 0 – Col 8

Pin Assignment in STM32

PB.0 – PB.7 (Row 0 – Row 7) PB.8 (Col 9)	PB.0 – PB.8 (Col 0 – Col 8)

STM32: Controlling DRAM by Software (Read Access)

Assume:
ColAdd = 0x03AB; (10 bits)
RowAdd = 0x00F7; (8 bits)
PC.2 = 1 ; /W
PC.0 = 1 ; /RAS high
PC.1 = 1 ; /CAS high



This code does an actual DRAM read access.

```
……
        ; 9-bit Row Decoder
GPIOB□ODR = ((ColAdd & 0x200) >> 1) | RowAdd;

GPIOC□ODR = 0x06     ; /RAS low
Delay();
GPIOC□ODR = 0x07     ; /RAS high
Delay();

        ; 9-bit Column Decoder
GPIOB□ODR = ColAdd & 0x1FF;

GPIOC□ODR = 0x05     ; /CAS low
Delay();
GPIOC□ODR = 0x07     ; /CAS high
Delay();

data = GPIOA□IDR     ; Get the data (byte)
```

PC.0 □ 1, PC.1 □ 1, PC.2 □ 1 = 111 = 07

PC.0 □ 1, PC.1 □ 0, PC.2 □ 1 = 101 = 05

STM32: Refreshing DRAM by Software

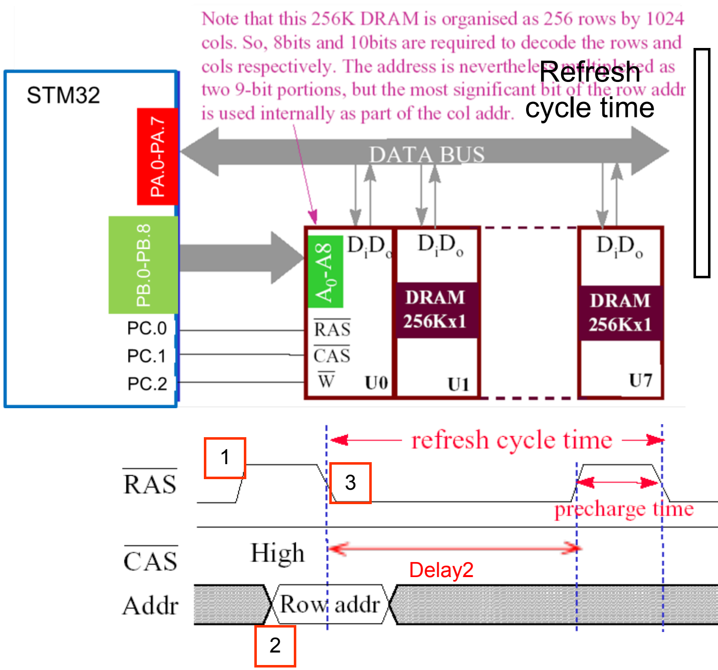
refreshing row by row

Assume:
PC.2 = 1 ; /W
PC.0 = 0 ; /RAS
PC.1 = 1 ; /CAS

This code refresh DRAM access.

```
……
i = 0;        ; Initialize refresh counter

GPIOC□ODR=0x07; /RAS high
GPIOB□ODR= i    ; Get next row address to refresh
Delay()         ; pre-charge time
GPIOC□ODR=0x06; /RAS low
Delay()         ;
i++             ; increment the refresh counter
……
……
```



Repeat this 256 times. This is written in-line rather than a loop to maximize speed

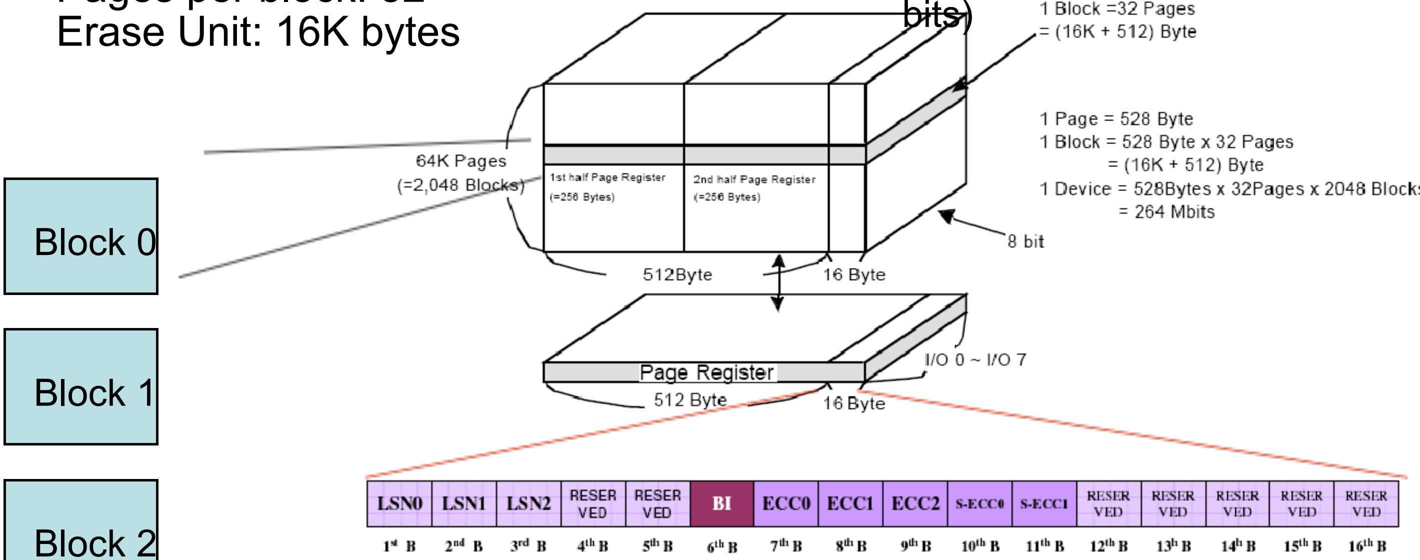
Flash Memory and its applications

	Applications	Performance	Type of Flash Memory
Code Storage  	Program storage for - Cellular Phone  - Switcher for telecommunications -PDA / POS / PCA -DVD  BIOS for - PC and peripherals	Important: - High speed random access - Byte programming Acceptable: - Slow programming - Slow erase	NOR (full address and data lines)
File Storage  	Memory Storage for - Digital camera  - Video camera - Voice recorder - Audio recorder  - PDA Mass storage for - Solid-State Disk  -- Hybrid HDD	Important: - High speed programming - High speed erasing - High speed serial read Acceptable: -Slow random access -I/O mapped access	NAND (Commands and data are multiplexed onto eight I/O lines)

Flash Organization

- First Generation of NAND flash
  - Page size: (512 + 16) bytes
  - Pages per block: 32
  - Erase Unit: 16K bytes

Samsung K9F5608X0C (32M x 8 bits)



> LSN : Logical Sector Number
> ECC0,ECC1,ECC2 : ECC code for Main area data
> S_ECC0,S_ECC1 : ECC code for LSN data
> BI : Bad block Information

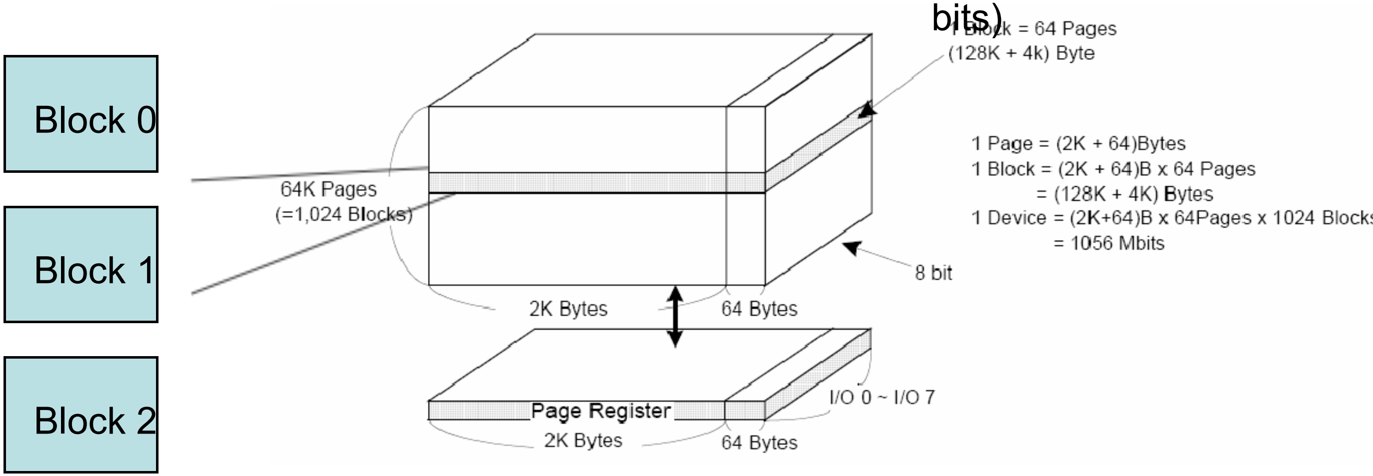
Sector: The smallest block of Flash that can be erased and/or programmed.

The logical sector number is converted to a real physical sector number of flash memory by some mapping Algorithm.

Flash Organization

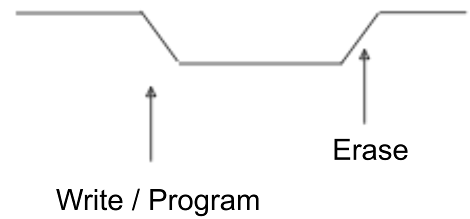
- Second Generation of NAND Flash
  - Page size: (2048 + 64) bytes
  - Pages per block: 64
  - Erase Unit: Blockwise Unit (128K bytes)

Samsung K9F1G08X0M (128M x 8 bits)



Flash Memory Characteristics

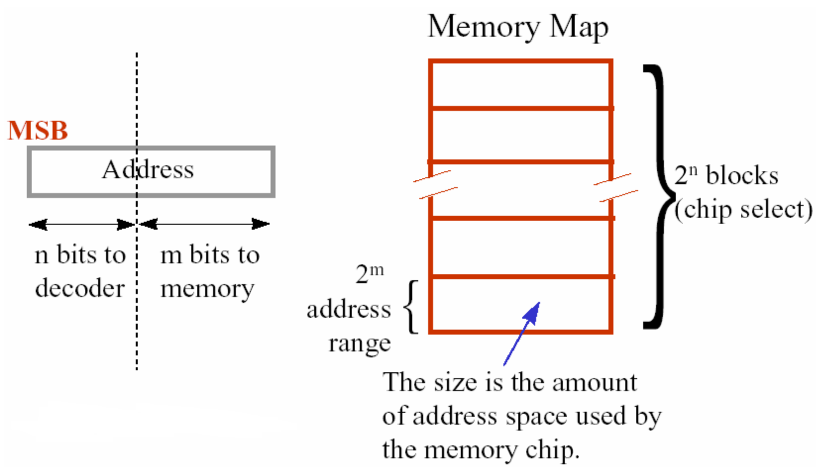
- Operations
  - Read
  - Write or Program : Change state from 1 to 0
  - Erase : Change state from 0 to 1



- Unit
  - Page (sector) : management or program unit
  - Block : erase unit

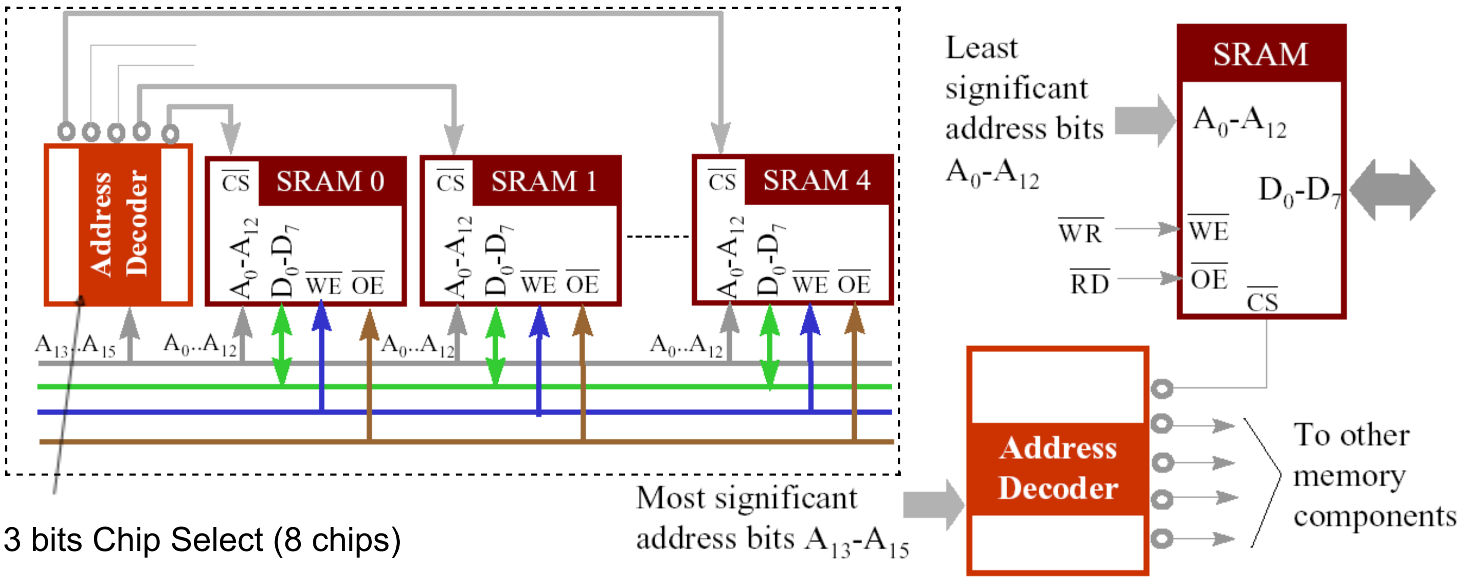
Paging in Memory Architecture

- There is an Address Space implied by the Address Lines on the bus
  - e.g. 16-bit addresses on a 16-line address bus implies up 216 addressable locations in the memory subsystem
- It is unusual to have one memory device spanning the whole address space
- Instead, there is a Memory Map which maps storage locations in particular memory components to addresses in the address space



The Memory Interfacing Task

- The bus connects to Memory Components via Memory-bus interface circuitry
- The tasks that this interface has to perform are
  - Address decoding (the principal task)
  - Matching bus control sign (RD/WR, PSEN, EA) to the control signals (RD/WR, OE, CE) of the memory components

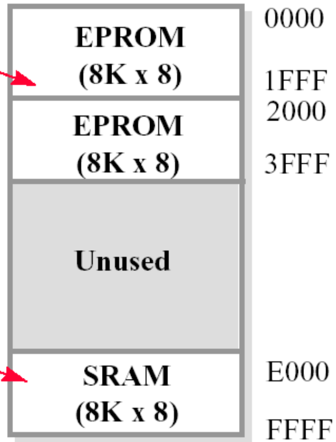


PSEN (Program Store Enable) is the read strobe for external instruction access.

Address Decoding: An example

1. Program codes
2. BIOS
3. Device drivers
4. ISRs
5. Operating System

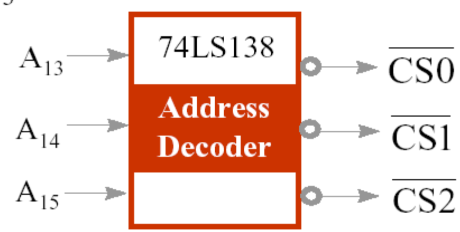
1. BIOS data area
2. Program data area



Bus Address	Device Type	$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$
0000 1FFF	8K EPROM	0	1	1
2000 3FFF	8K EPROM	1	0	1
4000 DFFF	unused	1	1	1
E000 FFFF	8K SRAM	1	1	0

The address decoder is a combinational logic to generate the Chip-Select signals for each memory component

Addr Range	A_{15}	A_{14}	A_{13}
0000-1FFF	0	0	0
2000-3FFF	0	0	1
E000-FFFF	1	1	1

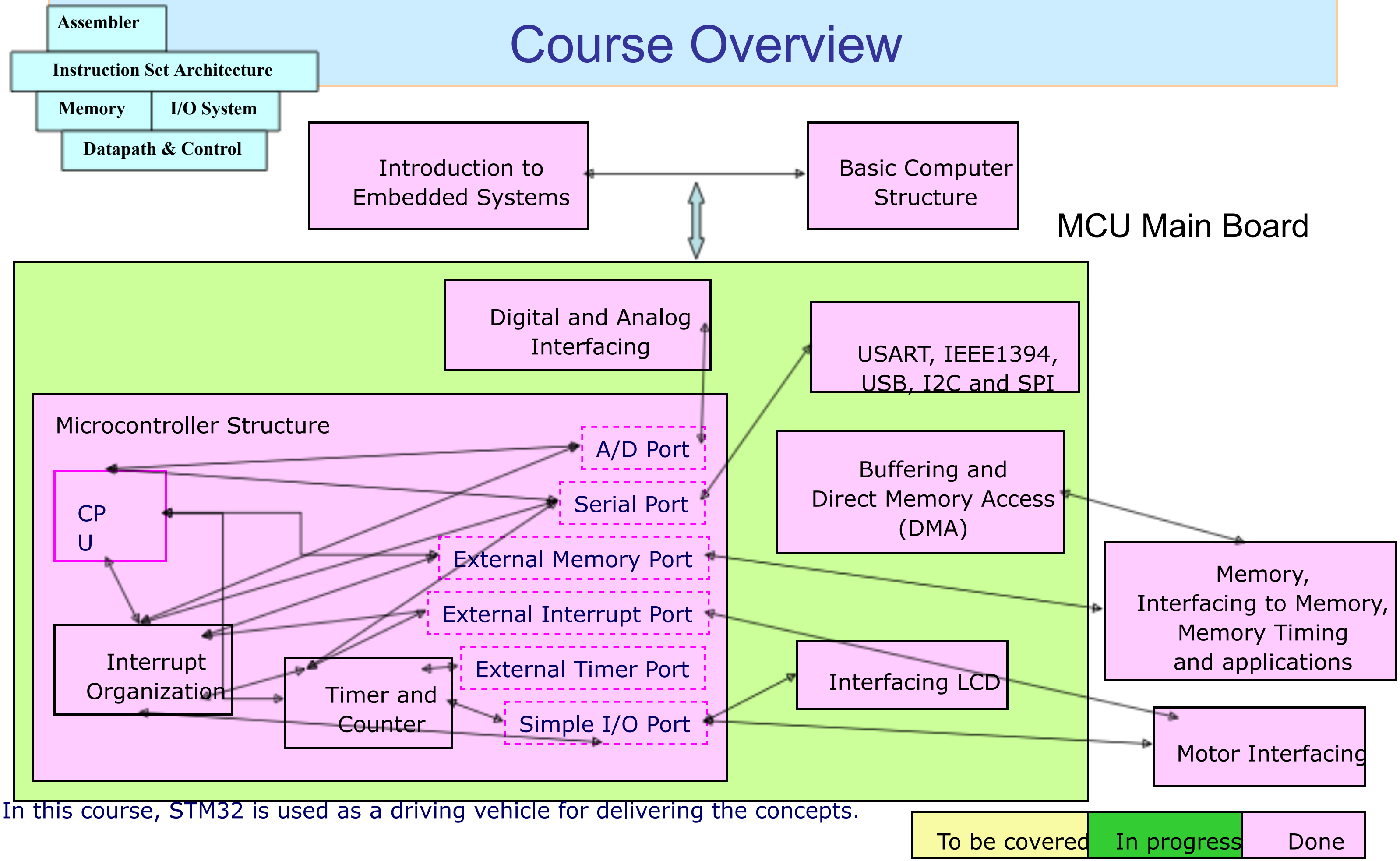


1110 0000 0000 0000 – 1111 1111 1111 1111

Reflection (Self-evaluation)

- Do you …
  - List several types of memory technologies ?
  - Design both hardware and software memory interfacing ?
  - Understand key considerations of memory architectures?
  - List upcoming memory architecture?

Course Overview



In this course, STM32 is used as a driving vehicle for delivering the concepts.