

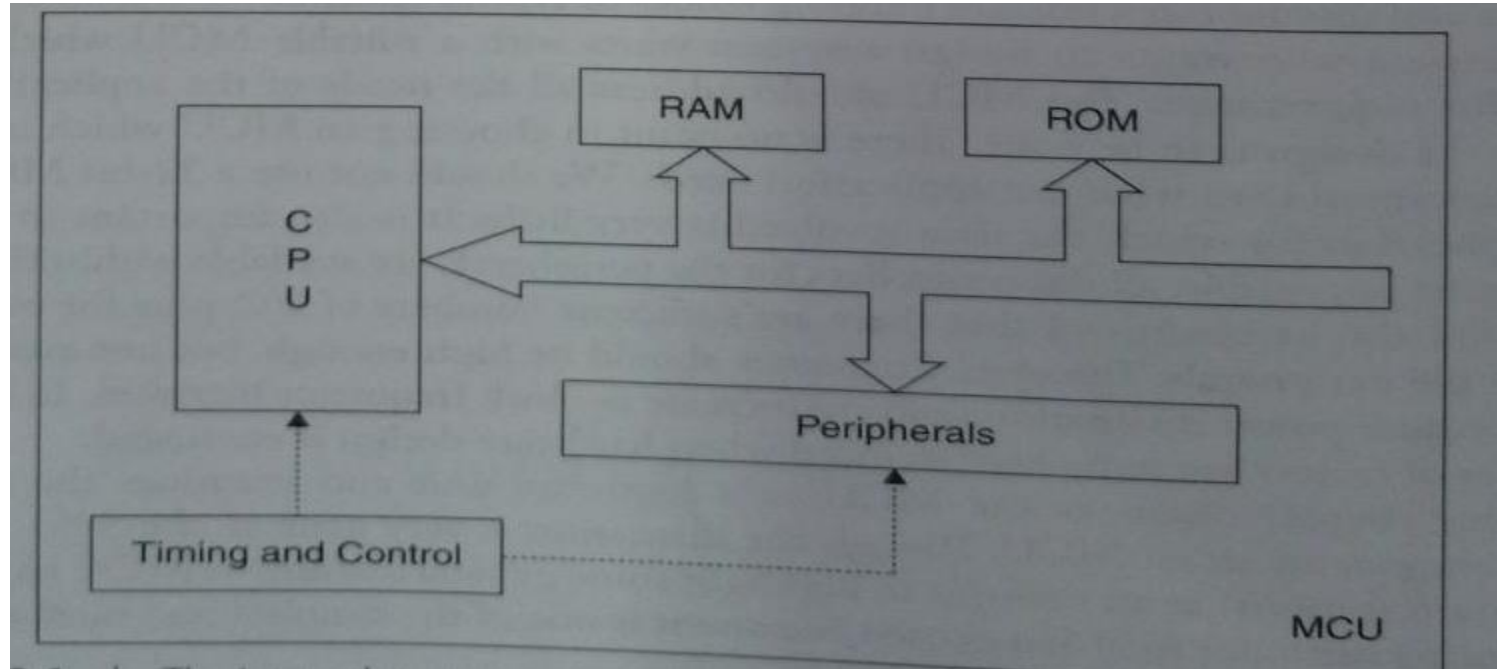
Embedded System – The Hardware Point of View

Chapter 2

Embedded System Design

UCS614

Microcontroller Unit (MCU)



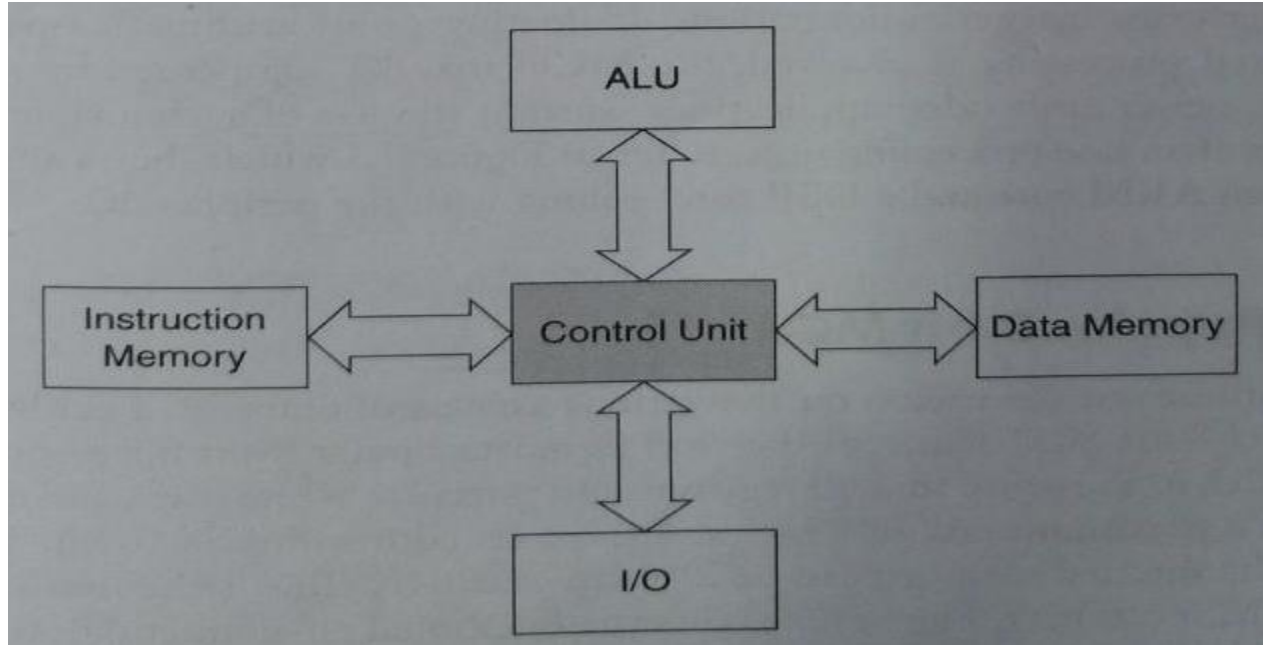
Microcontroller Unit (MCU)

- Processor or Computing engine
- Memory
 - RAM
 - ROM
- Peripheral interface
- Timing and Control

Processor: Architectural point of view

- Van Neumann architecture
 - Code and Data share the same address space
 - Both accessed by same bus
 - MPUs like x86 series
- Harvard architecture
 - Code and data are considered distinct
 - Accessed by separate bus
 - In MCU, it is achieved by instructions stored in ROM and data in RAM

Harvard architecture



Harvard architecture

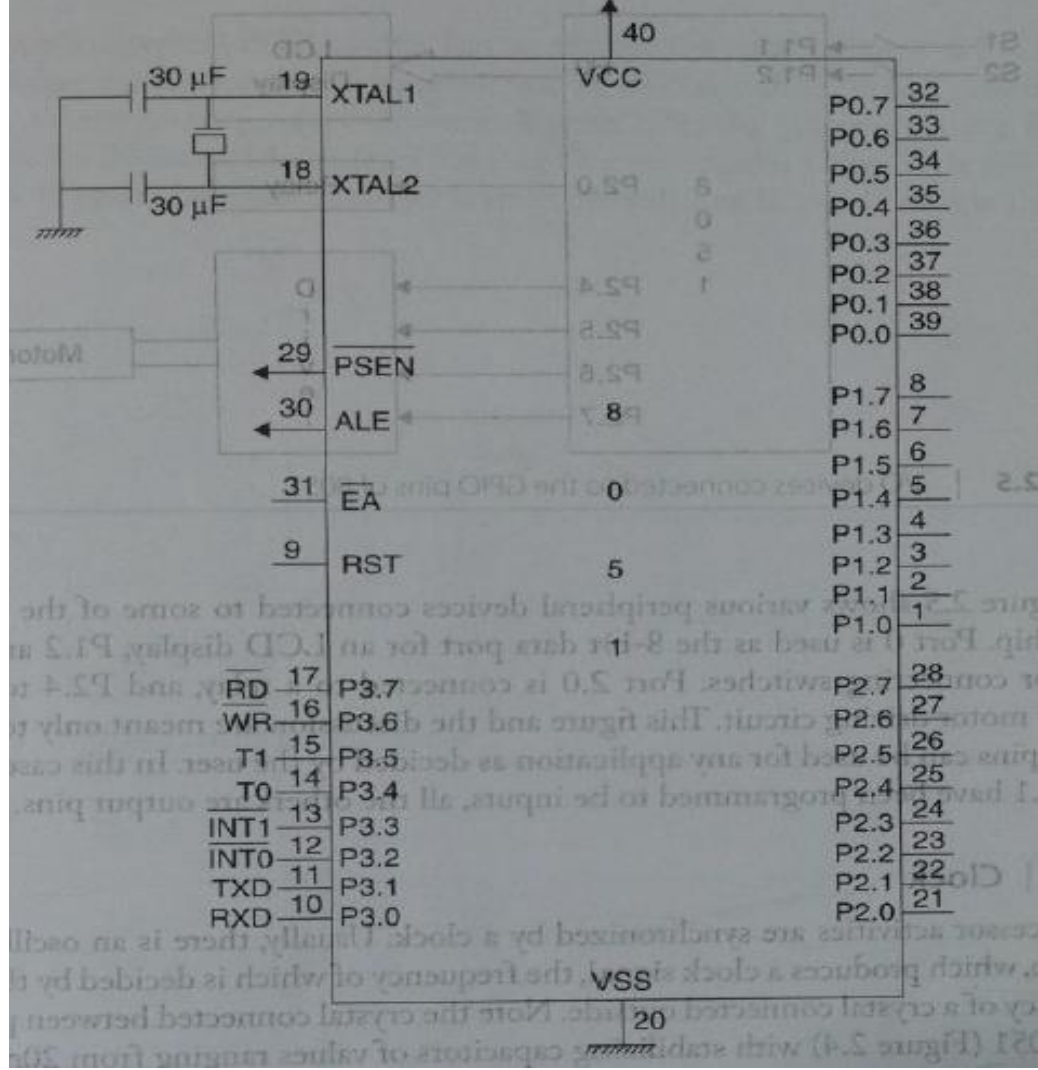
- MCUs are single chip computers
- RAM and ROM are inside the chip
- Buses access data and instruction are inside the chip

8051 MCU

- 8 bit MCU
- Data buses are 8 bits wide
- Address buses are 16 bits
 - Maximum 2^{16} bytes or 64K addresses
- RAM is relatively small and divided into
 - General purpose registers
 - R/W memory area
 - Special function registers (SFRs)

Important 8051 Features

- 64K bytes ROM
- 128 bytes RAM
- Four 8-bit I/O ports
- Two 16-bit timers
- Serial interface

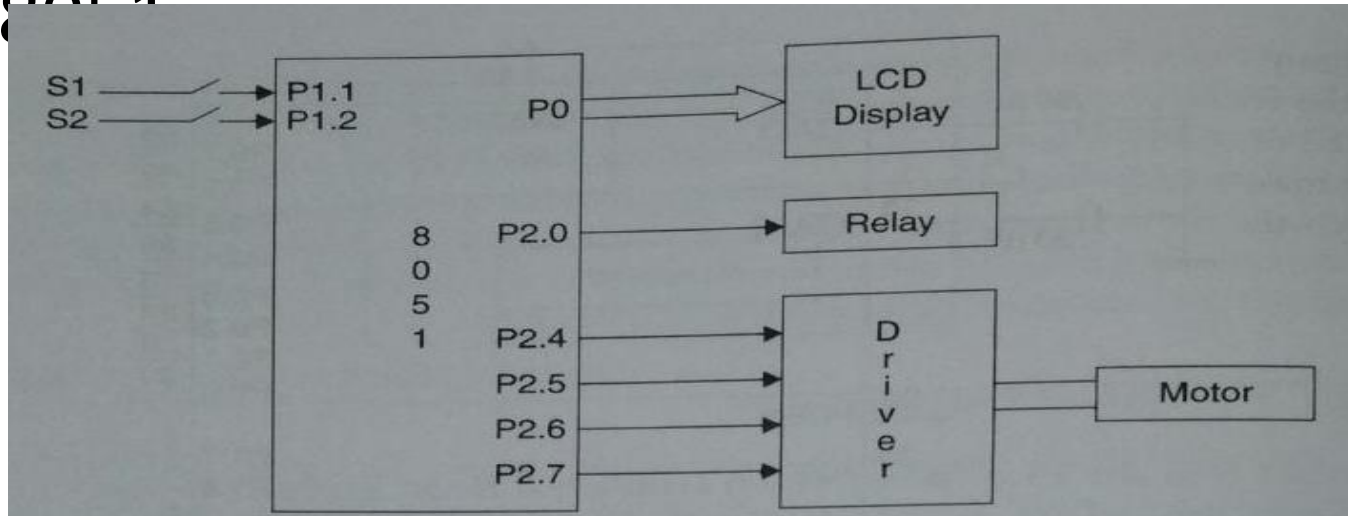


Pin Layout

- The 8051 is a 40 pin device, but out of these 40 pins, 32 are used for I/O.
- 24 of these are dual purpose, i.e. they can operate as I/O or a control line or as part of address or data bus.

General Purpose I/O (GPIO)

- 8 bits ports P0 to P3
- Example: I/O devices connected to GPIO pins of 8051



Port 0 and Port 1

- Port 0 is a dual purpose port, it is located from pin 32 to pin 39 (8 pins) and is labeled as AD0 to AD7.
- Port 1 is a dedicated I/O port from pin 1 to pin 8. It is generally used for interfacing to external device thus if you need to connect to switches or LEDs, you could make use of these 8 pins.

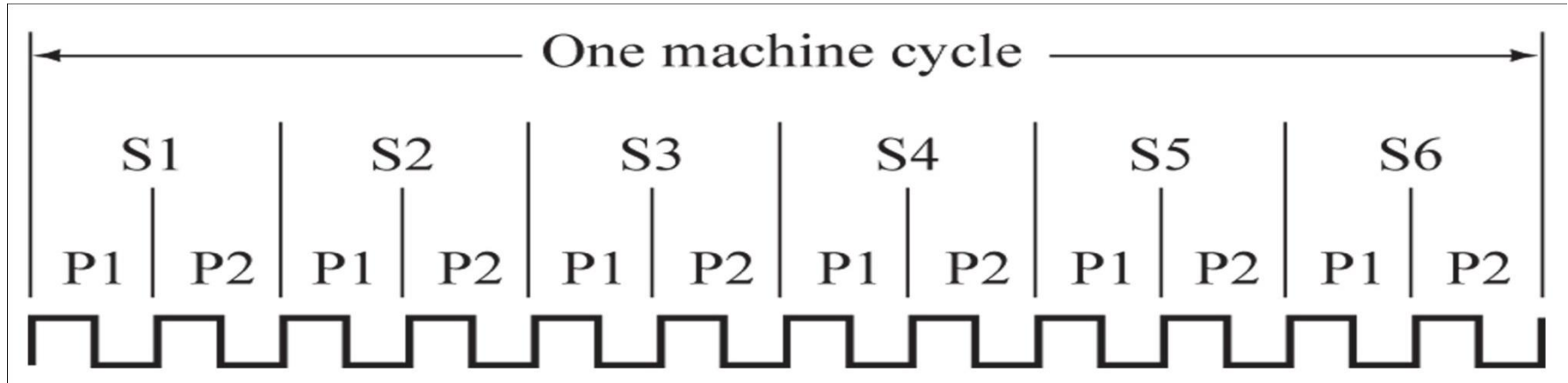
Port 2 and Port 3

- Like port 0, port 2 is a dual-purpose port. It can be used for general I/O or as the high byte of the address bus for designs with external code memory.
- Port 3 is also dual purpose but designers generally avoid using this port unnecessarily for I/O because the pins have alternate functions which are related to special features of the 8051. Indiscriminate use of these pins may interfere with the normal operation of the 8051.

Machine Cycle and Clock Cycle

- 12 clock cycles make one machine cycle.

Relationship between oscillator clock cycles, states, and the machine cycle



Memory Structure

- While most microprocessors implement a shared memory space for data and code (programs), microcontrollers has limited memory and the program is usually stored in ROM.
- In the 8051, both code and data may be internal but they are stored in separate memories, namely the internal ROM and RAM. Expandable to a max of 64K using external memory.

- PSEN (Program Store Enable)

- This is a dedicated control line on pin 29 and is used to enable external program (code) memory. This pin usually connects to an EPROM's Output Enable (OE) pin.
- This is a logic low pin as represented by the bar above the word PSEN, this means that during a fetch stage involving an instruction stored in external memory, the pin will be pulsed 'LOW'.

- ALE (Address Latch Enable)
 - This pin is used to demultiplex the address and data lines.
 - Remember that port 0 has 2 functions. As the low byte of the address bus and as the data bus. In designs with external memory, port 0 is connected to both the address and data lines of the external RAM thus during the part of the fetch cycle where the address is supplied, the ALE is pulsed to enable the G (gate) control pin of the latch IC thus the data goes to RAM and is interpreted as an address.

—

- EA (External Access)
 - If you need to connect to external ROM then this pin must be tied LOW (0V).
 - This pin must be tied high (+5V) if the programs executes from internal ROM.

RST (Reset)

- This is pin 9 of the IC and is used as the master reset for the 8051. In order for the 8051 to recognize that a reset has occurred, this pin must be brought HIGH for at least two machine cycles. During normal operation, this pin must be at logic LOW.

- Power Connections
 - The 8051 requires a +5V input on its Vcc input (pin 40) and Vss connection is on page 20.

Accessing External Code Memory

- If the design involves external code memory, both P0 and P2 should not be used as general purpose I/O since P2 is now the Higher address bus while P0 is the multiplexed Lower address bus and the data bus.

- As stated earlier, the PSEN pin must be used.

FIGURE 2-8 Multiplexing the address bus (low-byte) and data bus

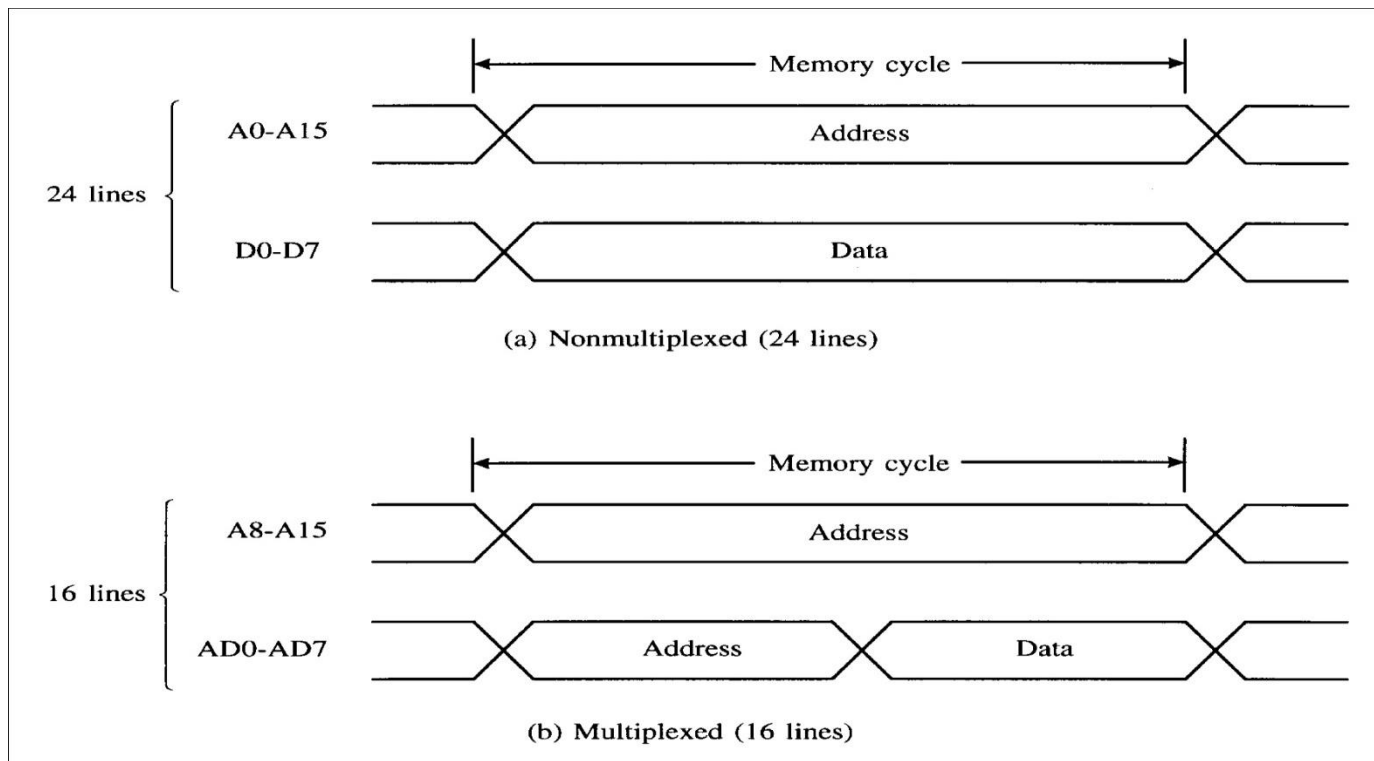
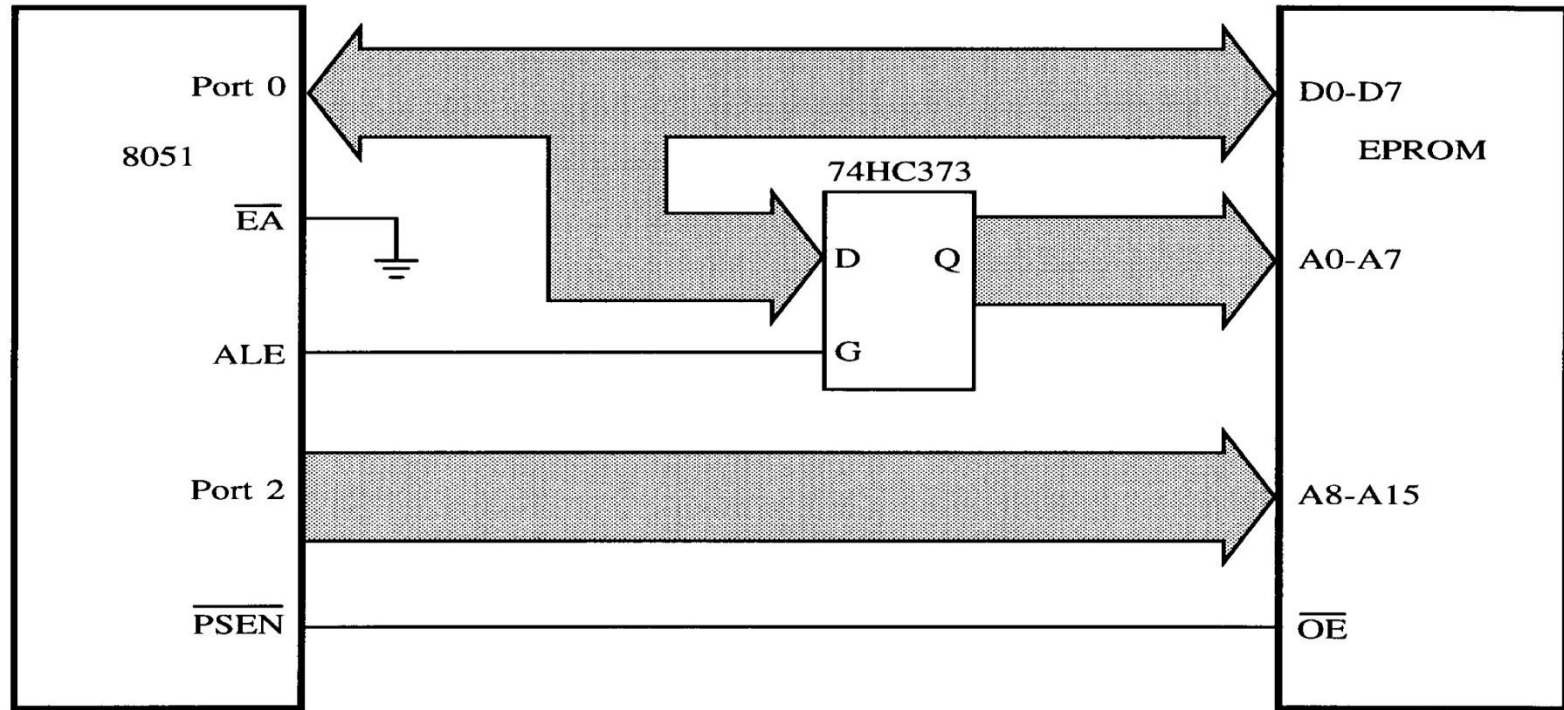


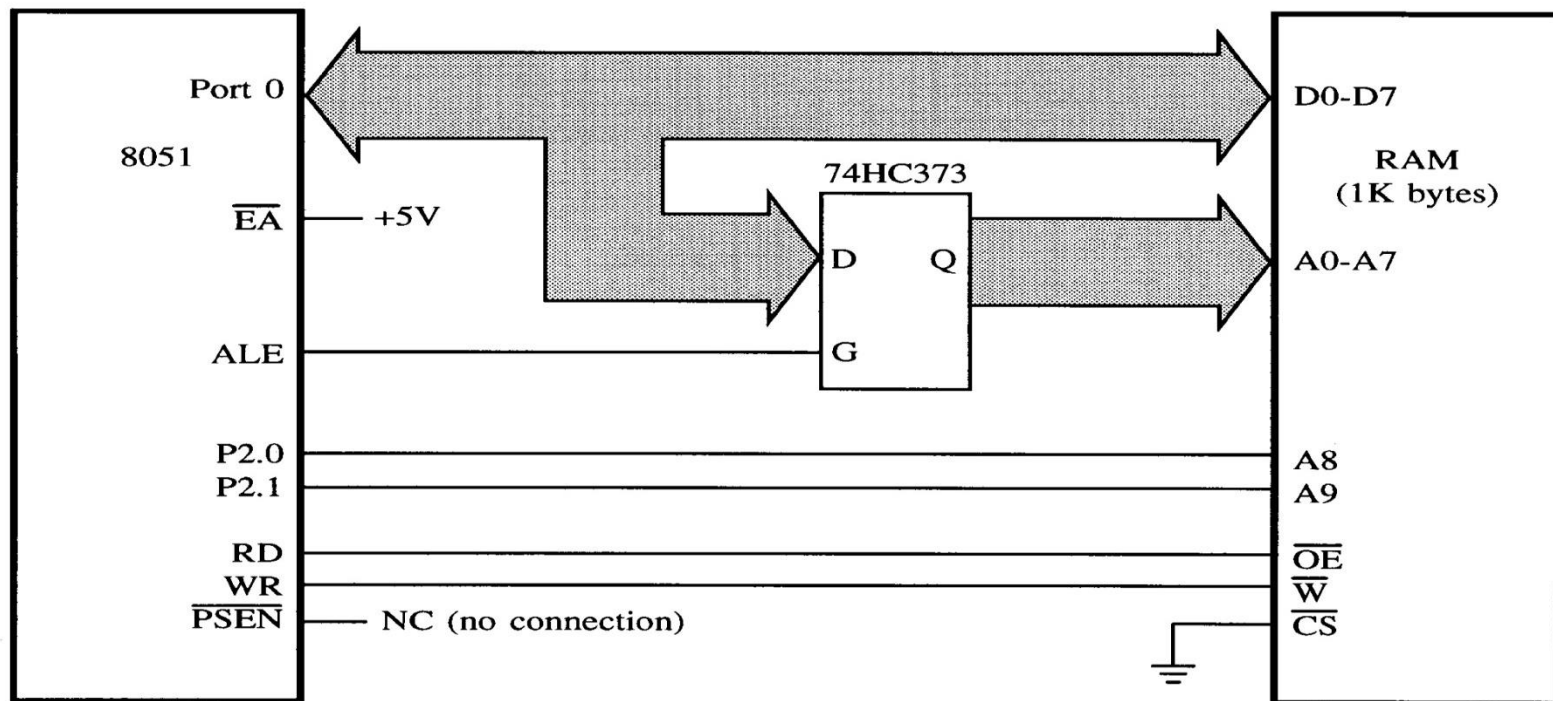
FIGURE 2-9 Accessing external code memory



Accessing External RAM

- For designs with external RAM, a typical connection is shown in fig.2-12. Note the control lines that must be used.

FIGURE 2-12 Interface to 1K RAM



Clock

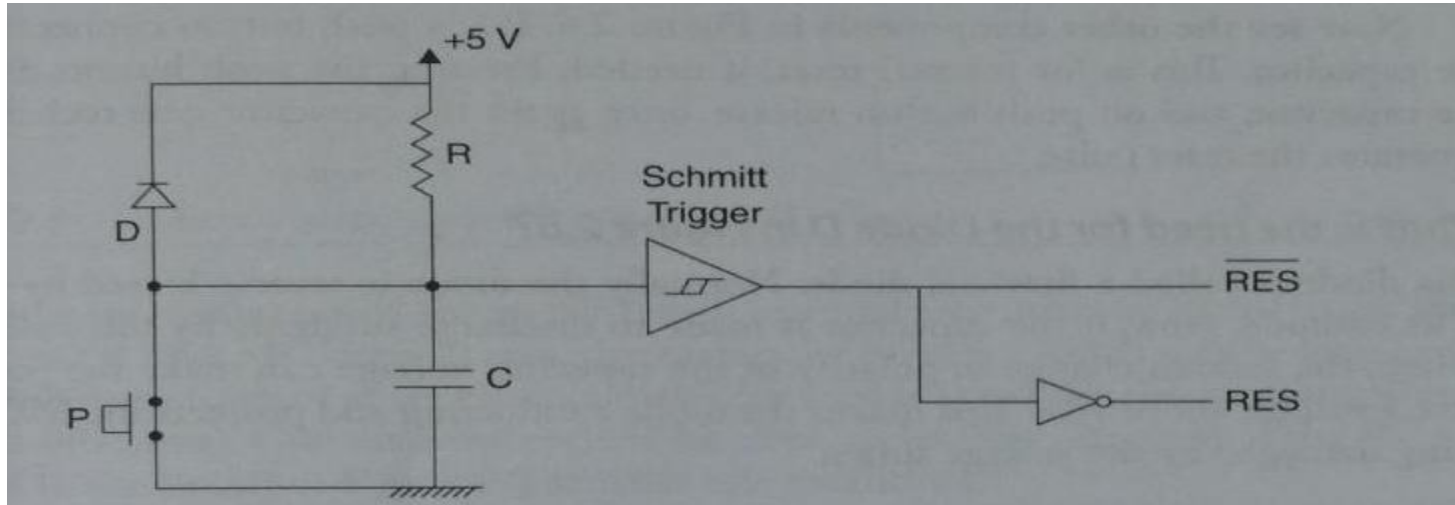
- All processors activities are synchronized by a clock.
- Oscillator inside the chip produces clock signal
- Pin 18 and 19 of 8051 connect to crystal
- Speed of computation is directly related to the clock frequency
- Each MCU can support up to certain maximum frequency of crystal
- Timing calculations of timers, counters etc are based on frequency of clock

Power on Reset

- When the power is switched on, processor is reset
 - Internal registers get reset
- MCU needs specific pulse for reset
 - Pulse is either high or low
 - Needs to have minimum period
- Reset pulse should appear once, when power is switched on
- “Power on reset” circuit is to be provided externally

Power on Reset

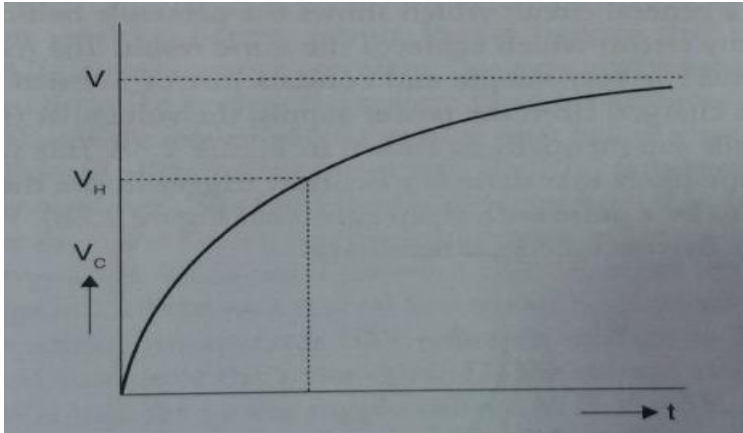
Power on reset circuit:



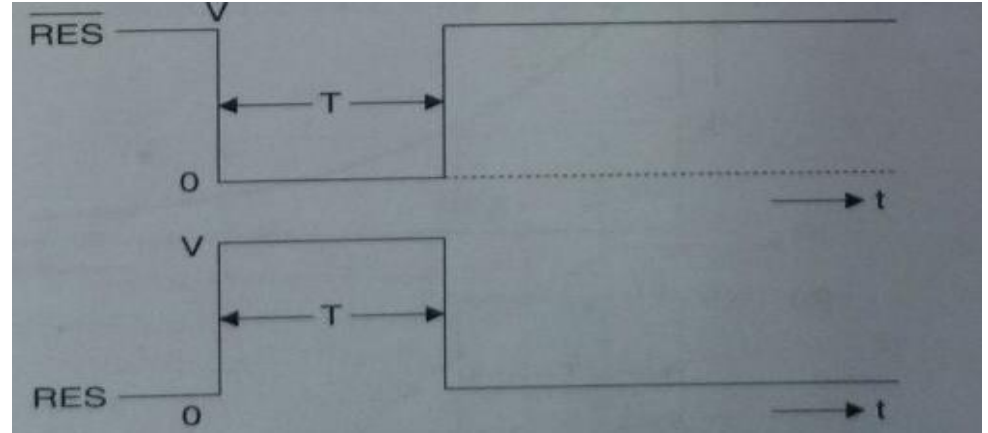
Power on Reset

(a) Voltage waveform across the capacitor:

(b) Pulse at RES and RES



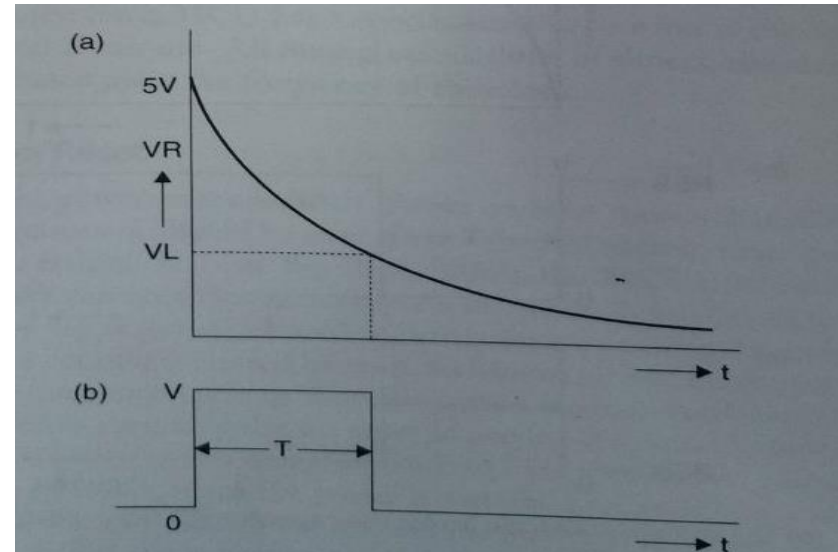
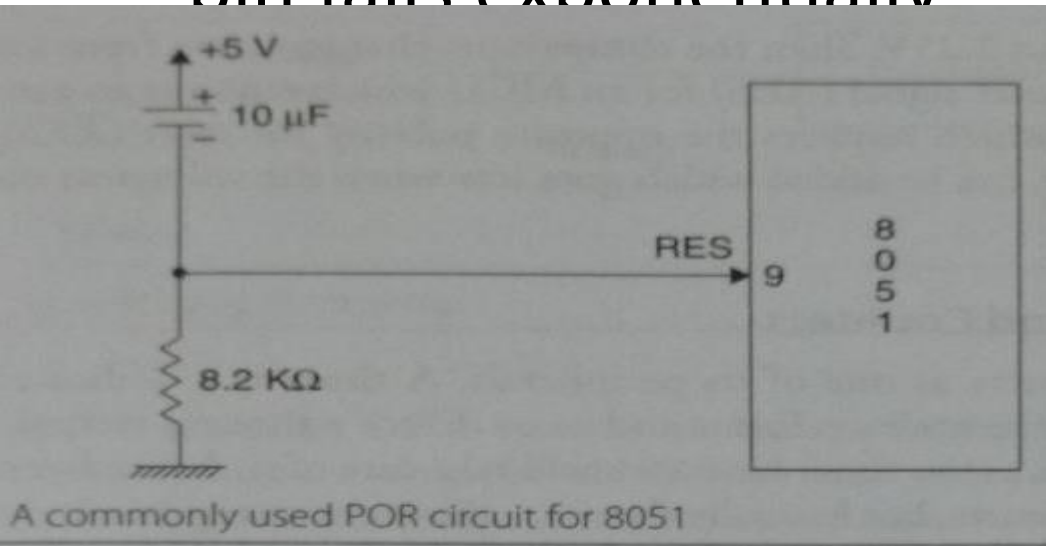
(a)



(b)

Power on Reset

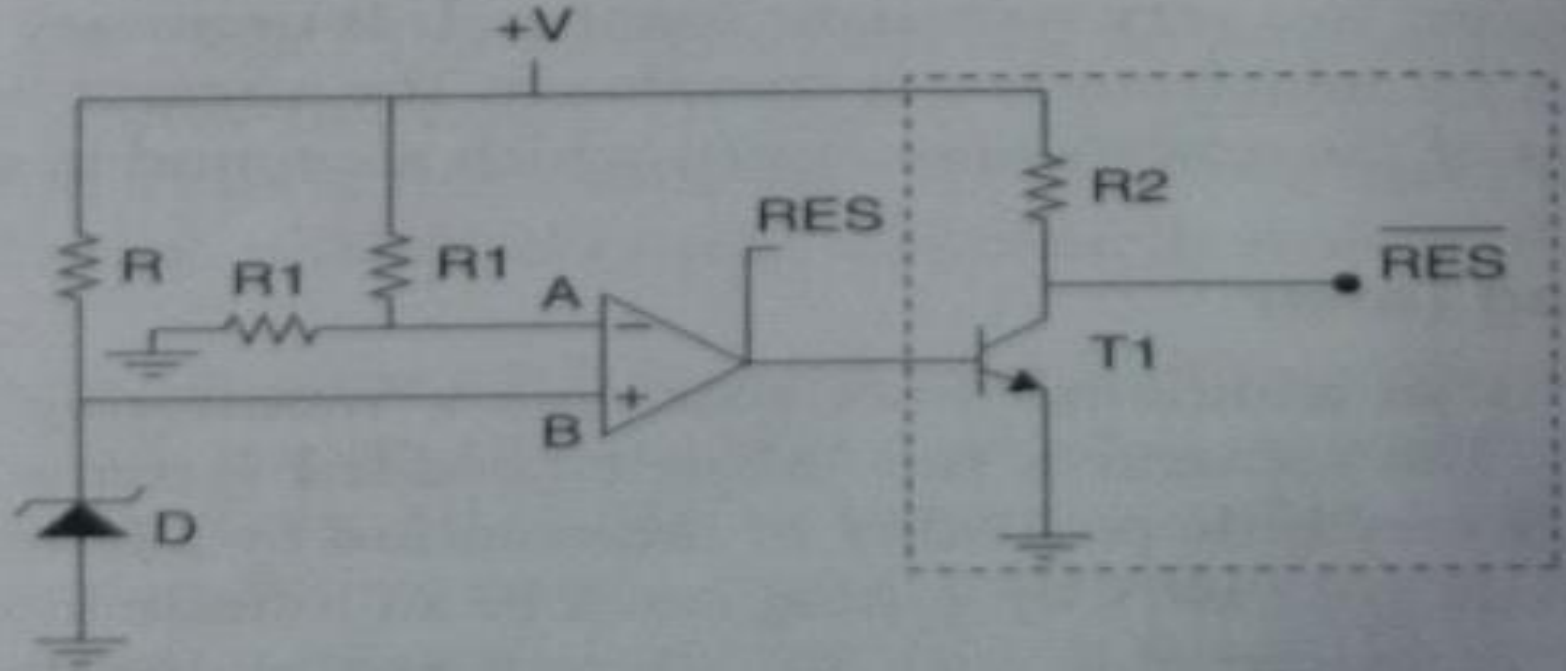
- In 8051 POR consist of resistor and capacitor
- As capacitor gets charged, the voltage at reset pin falls exponentially



Brown Out Reset

- When battery gets drained, MCU may not work correctly
 - Wrong decision and actions
- To protect the system from such situation
 - It is ensured that if power supply voltage goes below a specified level, the MCU should be reset
- The 8051 family does not have “Brown OUT Reset” feature

Brown out reset



A simple circuit for brown out reset

Timers and Counters

Timer

- MCUs have timers as peripherals
- Timer is a dedicated hardware for 'timing' events
 - Processing unit does not interfere, once timer is started (H/W will take care of it)
- Timer count register and mode register associated (Other registers may also be associated)

Timers and Counters

- Processor clock works as reference clock
- Clock frequency can be divided to get lower frequency clock

Working of Timer

- Number is loaded in timer count register
- Timer is started
- Counter keeps increasing or decreasing until it reaches maximum or minimum value
- At this point, count register reset to 0
- Interrupt or flag bit set to indicate this

Timers and Counters

Counter

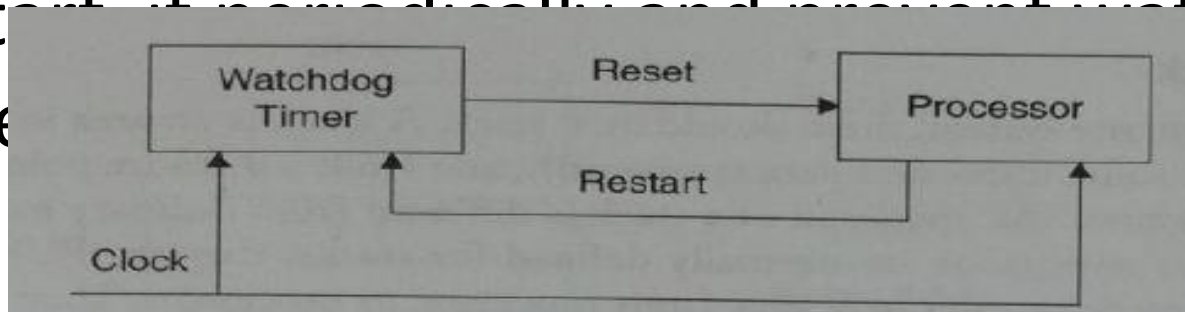
- Counts external events
- Generally does not use processor clock
- Count register associated with counter gets incremented for each occurrence of event under consideration

Watchdog Timer

- Does a 'monitoring' job and resets the system, if necessary
- Fulfil the characteristics of 'self-reliant' in embedded system
 - Reduces human intervention in case associated software get stuck in infinite loop
- Similar to any other timer

Watchdog Timer

- Loaded with a 'count' which decrements down to 0
- If reaches to 0, resets the processor.
- If system is doing its job correctly, s/w will restart it periodically and prevent watchdog time



Real-time Clock

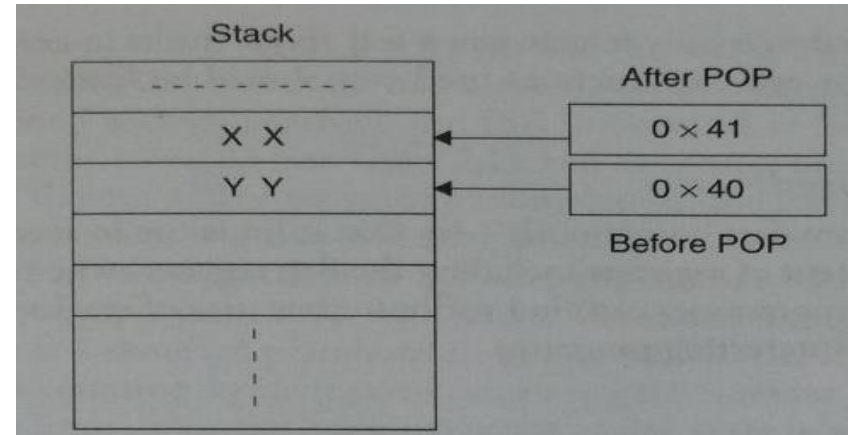
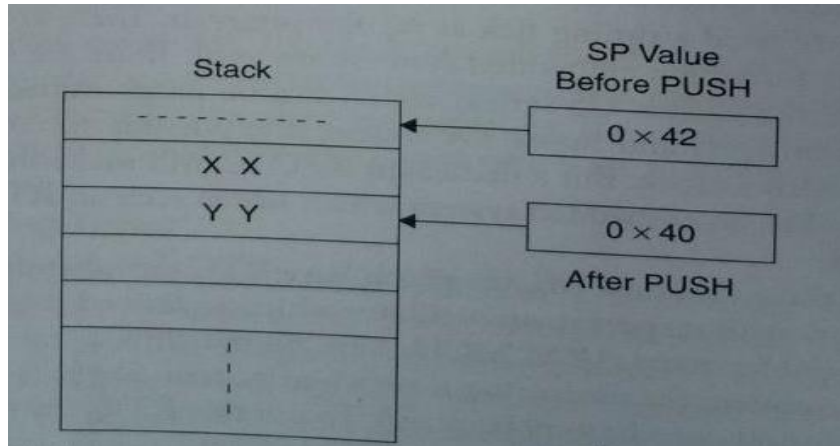
- Refers to the 'real' time

Stack

- Last In First Out (LIFO)
 - PUSH and POP operation
- Ascending stack
 - Starts from low memory address
 - On PUSH operation, item stored in higher memory address
- Descending stack
 - Grows downwards

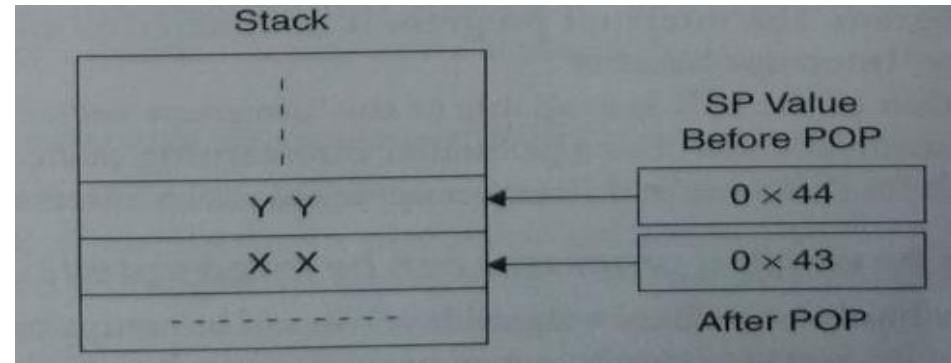
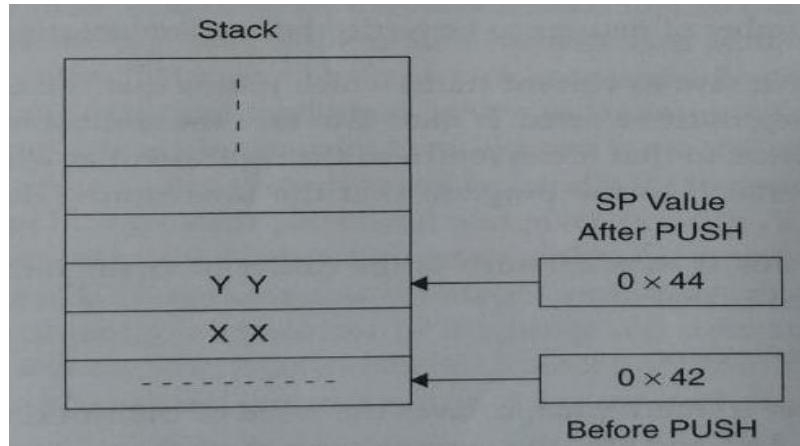
Stack

PUSH and POP operation in descending stack



Stack

PUSH and POP operation in ascending stack



Interrupts

- For executing more important activity by suspending current activity in execution
- In case of interrupt, processor completes the instruction it is currently running
- Takes up the new task
- Once interrupting task is over, return and continue execution previous task

Interrupts

Key points

- Processor must save its current status
 - Content of registers (in stack), address of next instruction (value of program counter)
- ‘Branch’ to the program for which interrupt was generated (Instructions of interrupt program designated as ‘Interrupt Service Routine’)
 - Interrupt vector stores address of first instruction of ISR
- After execution, ‘return’ and resume execution where it had left

Interrupts

What are the ways by which a processor can be interrupted?

- Done by hardware (activate interrupt pin of the processor e.g. pin 12 of 8051)
- Using MCUs inbuilt peripherals
 - Interrupt by timer
- Software interrupts (write instructions in program)
 - Mostly applicable in MPUs

Interrupt structure of 8051

- Five interrupts excluding reset

Sl. No.	Interrupt	Vector
1	Reset	0×000
2	$\overline{INT0}$	0×0003
3	Timer 0	$0 \times 000B$
4	$\overline{INT1}$	0×0013
5	Timer1	$0 \times 001B$
6	Serial R/T	0×0023

Interrupt structure of 8051

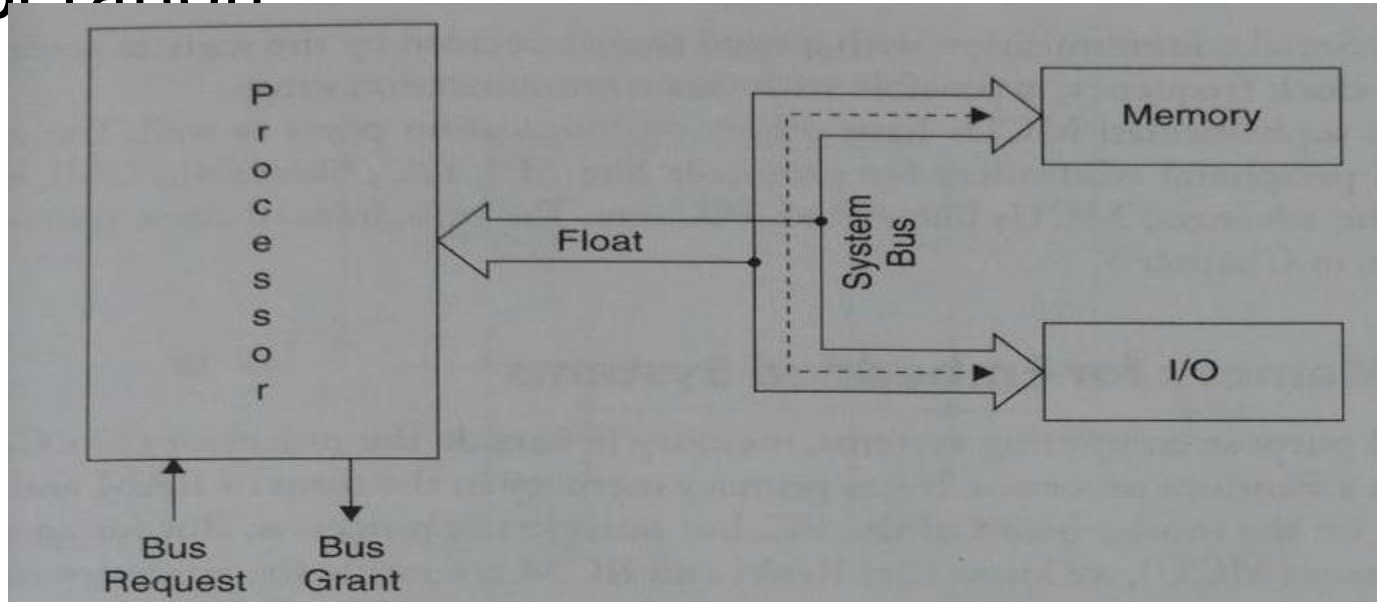
- Two external hardware pins INT0 and INT1
 - Pin 12 and 13
- Two timer interrupts Timer0 and Timer1
- Serial communication interrupt
 - Serial transmission as well as reception
- Except reset all interrupts can be enable and disable as per the user need.

Direct Memory Access

- To send data in memory to I/O or vice versa
 - Print a large chunk of data
 - No involvement of the processor
- To isolate processor from this process
 - Connection to memory and I/O is blocked
 - Buses of the processor are to be tri-stated (High Impedance state)

Direct Memory Access

Processor buses in float state during DMA operation



Communication Ports

- MCUs have UART as standard peripheral
 - Registers, Pins and interrupts associated with this peripheral
- In 8051 pins 10 and 11 for serial reception and transmission

Memory for Embedded System

- Random Access Memory
 - Static RAM
 - Dynamic RAM
 - SD RAM
- Read Only Memory

Introduction

- RAM and ROM inside processor chip.
- Size of memories varies chip to chip.
- If requirements of memory are more than the available on chip, external memory can be added.

Semiconductor Memory

- Data is stored in memory and it is usually defined as byte oriented i.e. One address corresponds to one byte in the memory .
- When one address is accessed one byte is read or written into it.
- For getting a word of four bytes, four consecutive locations with four addresses have to be accessed.

Semiconductor Memory

- Reading and writing takes some amount of time which is termed as **memory access time**.
- For reading, it is time when address is placed at address pins and data is available at the data pins.
- **Memory cycle time** is the time interval between two consecutive memory accesses.

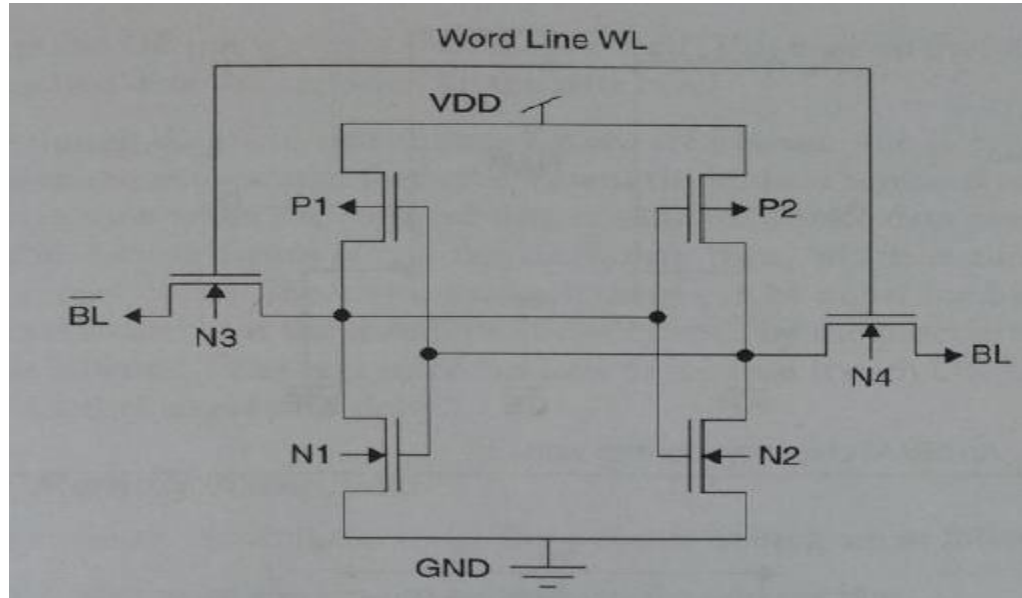
RAM

- RAM –**Random Access Memory**
- In which any location can be accessed randomly with same latency.
- It is **volatile memory** -power removed, data lost.
- Fastest and most expensive RAM is SRAM i.e. Static RAM.

SRAM

- Each cell holds either 0 or 1.
- This content is static, because content is stored as voltage which does not change with time.
- Memory is realized using MOSFETs and the most commonly used type of memory cell needs six transistors and is called 6T cell.

SRAM Memory Cell



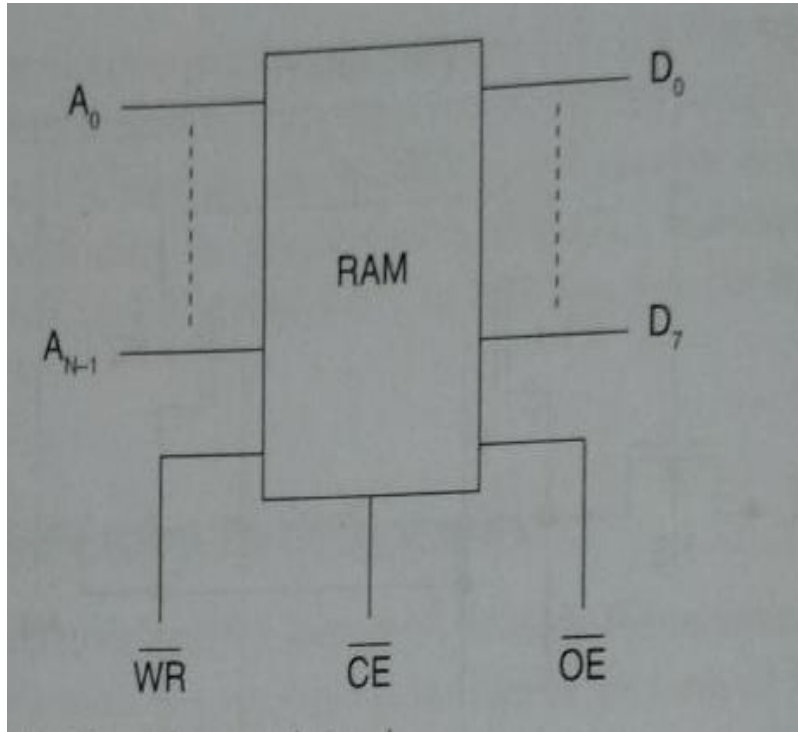
SRAM Memory Cell

- It has two cross coupled NMOS transistors (N1 and N2) acting as bistable multivibrator and two PMOS transistors (P1 and P2) acting as their loads.
- The access transistors N3 and N4, and the wordline WL and bitlines BL and Blbar are used to read and write to and from the cell.

SRAM Memory Cell

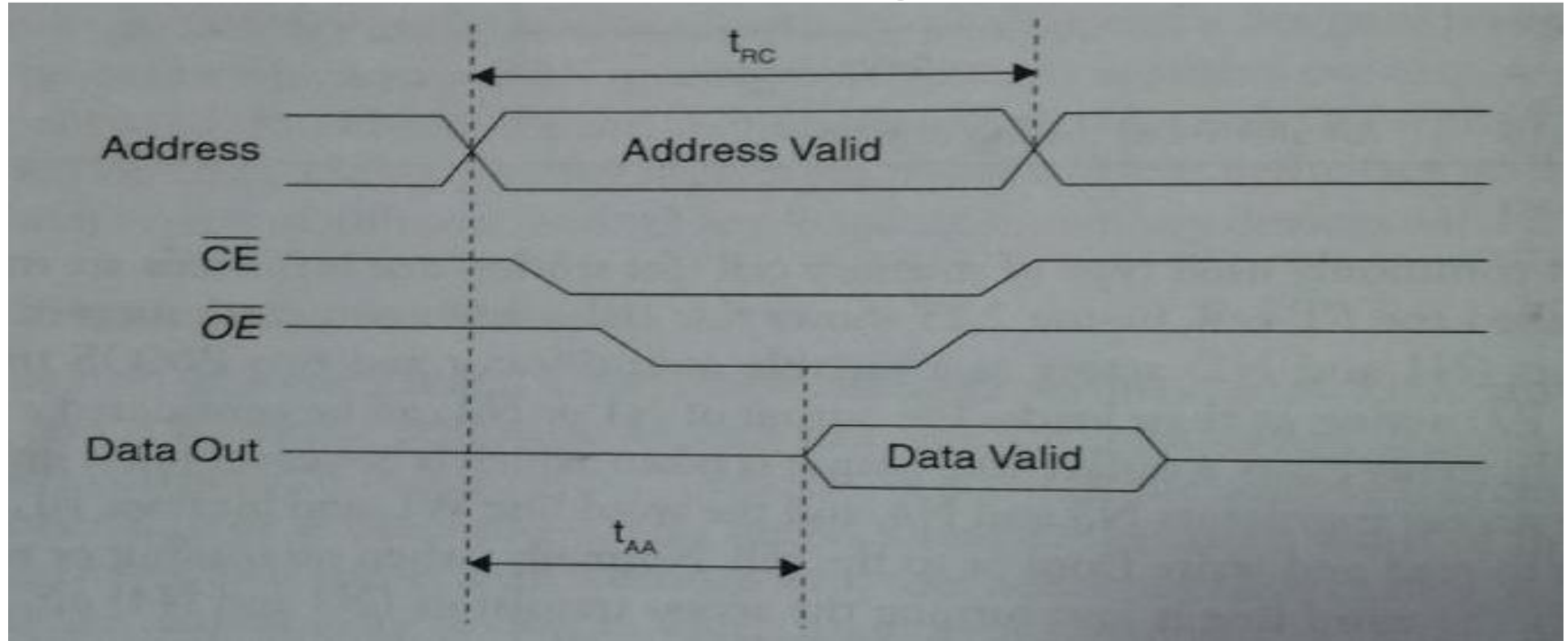
- When no reading or writing is required, word line is low, turning N3 and N4 off.
- To write data into the cell, logic data is placed on the bit line and complementary on inverse bit line.
- Then access transistors are turned ON by setting word line to high.
- For reading, the word line is turned ON to activate the access transistor while the information is sensed at bitlines.

SRAM Chip



- N address lines, 8 data lines and control signals for reading and writing.
- \overline{CE} line is activated, chip become usable.
- \overline{WR} is write control signal and
- \overline{OE} is read control signal signals.
- It enables the data lines for reading

SRAM Asynchronous Read timing



SRAM Read cycle

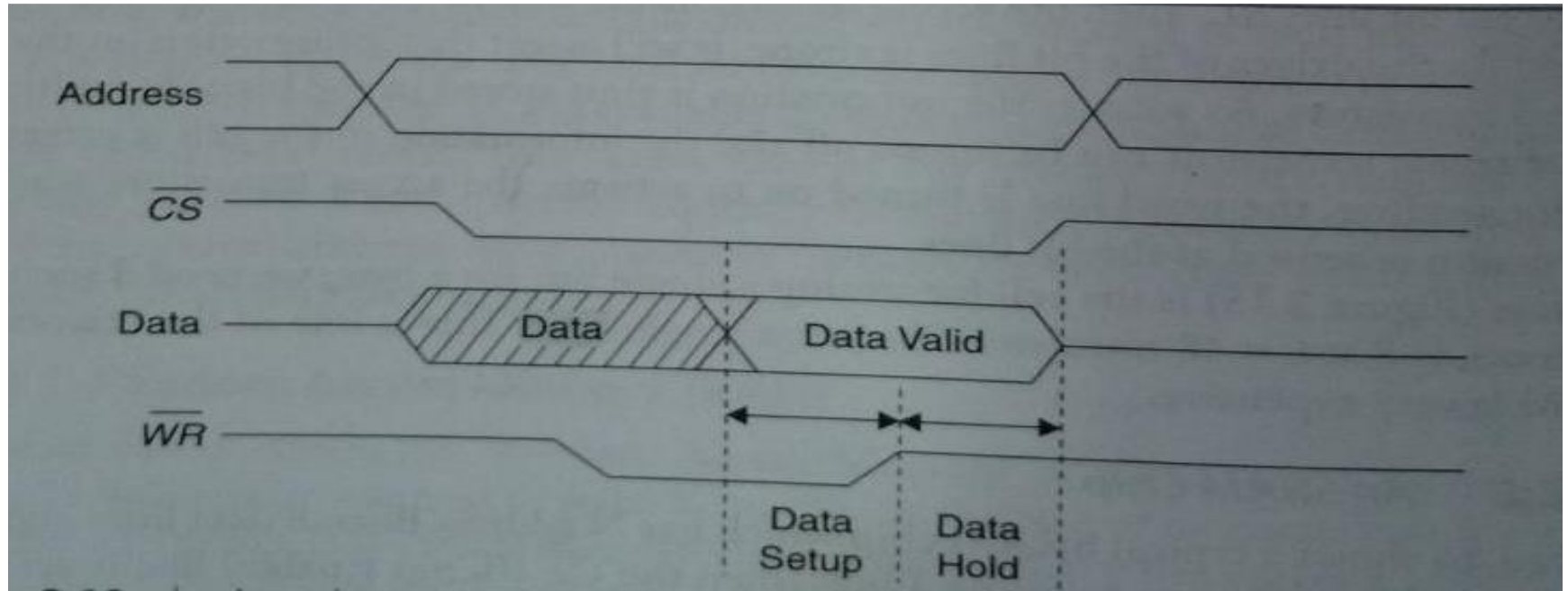
1. Place the address of byte to be read, on the address bus.
2. Ensure chip is activated by making \overline{CE} low.
3. Activate \overline{OE} pin which is \overline{RD} pin itself. This ensures that data is read.
4. The required data then appears on the data bus.

SRAM Read cycle

- **Read Access time:** this is the time measured from the instant the address is placed on the address bus to the point in time when the required data is available on the data pins. TAA shown in diag.
- **Read cycle time:** minimum time between two read cycles.

These can be equal for SRAM as data is available at data pins, it can be read by processor and a new read cycle can be initiated.

Asynchronous write Timing



Memory write cycle

1. Place the address of byte to be written on the address bus.
2. Ensure that the chip is activated by making \overline{CE} low.
3. Place the data to be written on the data bus.
4. Activate the \overline{WR} line. Only then the data is considered to be valid
5. The data is then written into the addressed location.

Merits and Demerits of SRAM

Merits

- CMOS is used. It uses less power than DRAM
- They are as fast as CPU because of using the same technology as the CPU.
- Any MCU will have a certain amount of SRAM inside it by the name on chip RAM.
- For many MCUs this SRAM includes the internal registers which are arranged as banks.

Demerits

- At higher frequencies, it can consume significant amount of power.
- Quite Expensive.

Synchronous SRAM(SSRAM)

- Ordinary SRAMs are asynchronous as there is no clock signal for timing the read and write operations.
- In such synchronous SRAM, processor clocks create the timing for read and write.
- SSRAMs are used in high speed applications.
- They are used as cache for power PC and Pentium –based workstations.

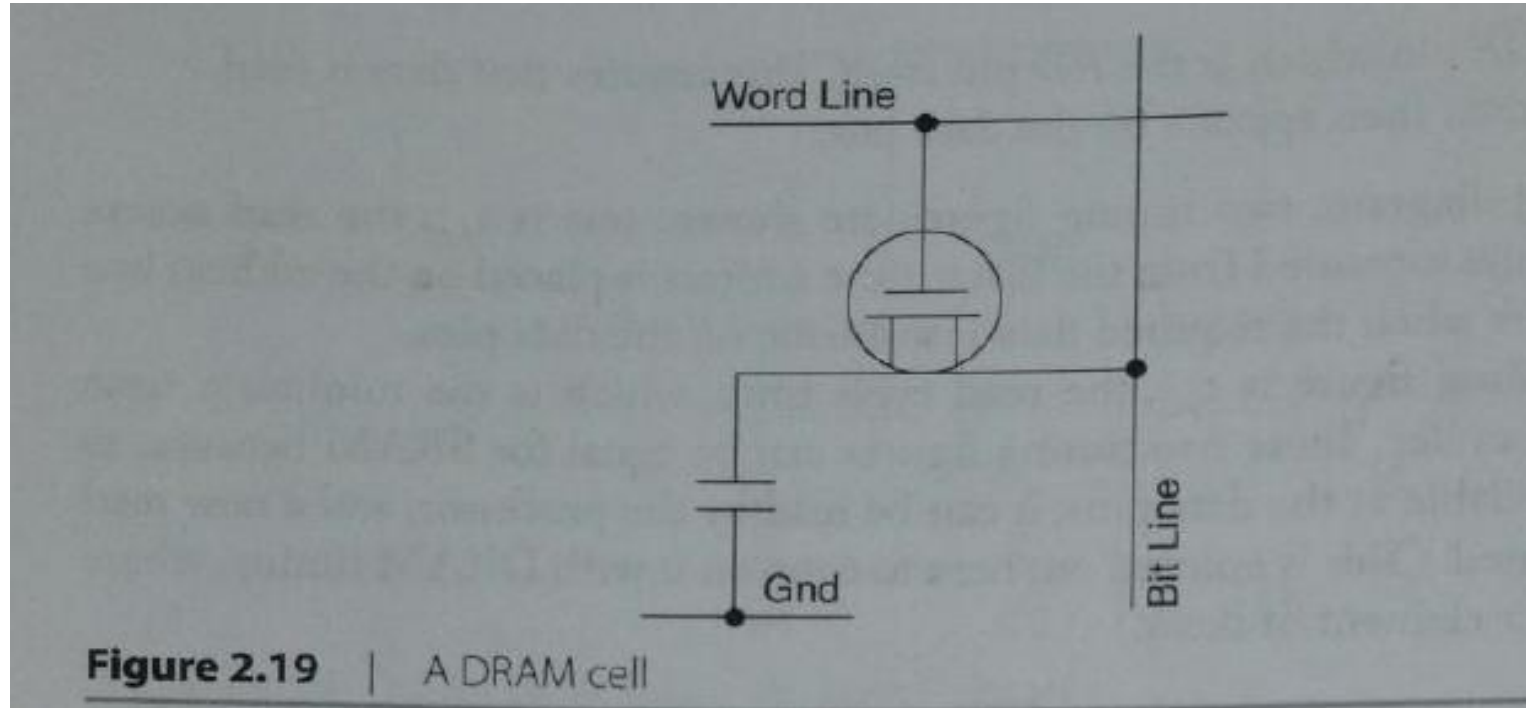
Dynamic RAM(DRAM)

- It is designated as dynamic because its content does not remain unchanged or static as in SRAM and hence frequent refreshing is necessary.
- A DRAM memory cell consists of a single field effect transistor(FET) and a capacitor.
- Amount of charge stored in the capacitor that decides whether the cell stores 1 or 0.

Dynamic RAM(DRAM)

- Capacitor does not hold charge indefinitely as the charge in a capacitor 'leaks off' and needs to be replenished.
- This action of replenishing the charge that gets lost is done by **refreshing** the cell at regular intervals.

DRAM Memory Cell



Memory Control of DRAM

- Word line and read line are connected as shown to select the required bit within memory to be read or written to.
- Multitudes of Such cells form word consisting of bits
- Memory addresses are decoded and converted as rows and columns of matrix that memory elements are arranged in

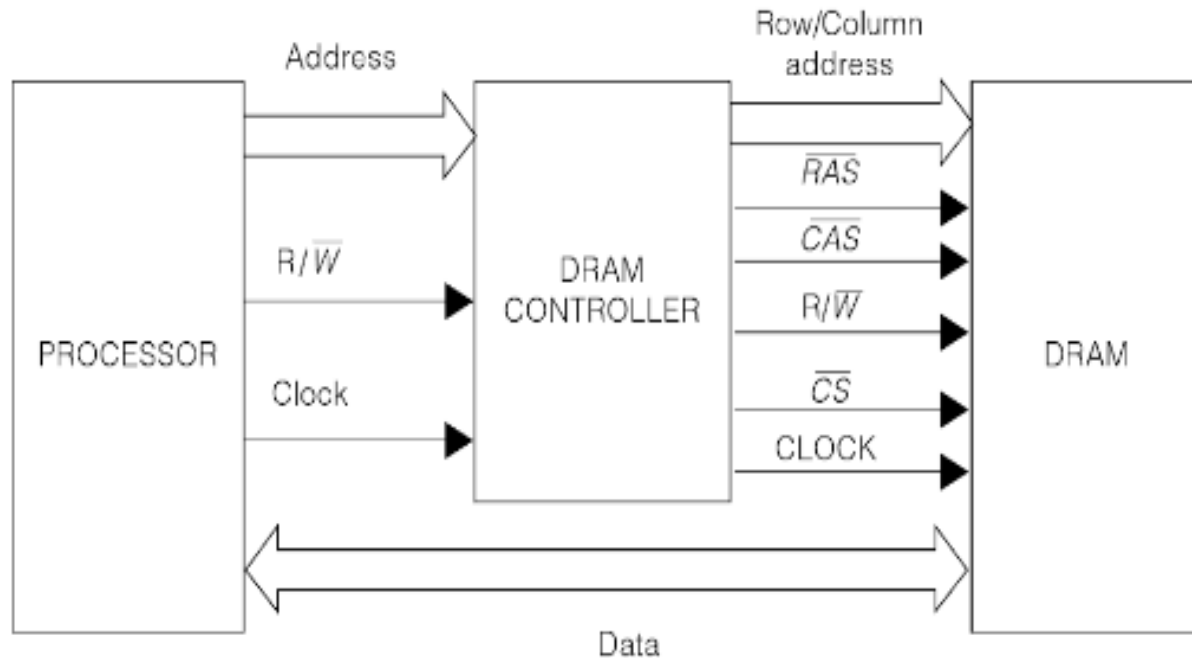
Read Cycle of DRAM

- Processor when address memory sends the complete address on its address pins.
- Between processor and DRAM chip there is a memory controller whose function is to split the address into two as columns and rows

Memory Control of DRAM

- A DRAM has only half the no. of address [ins as the address supplied by processor, because the address pins of DRAM chip are multiplexed in time for row and column addresses.
- The memory controller should also generate the signals necessary for reading or writing to DRAM.

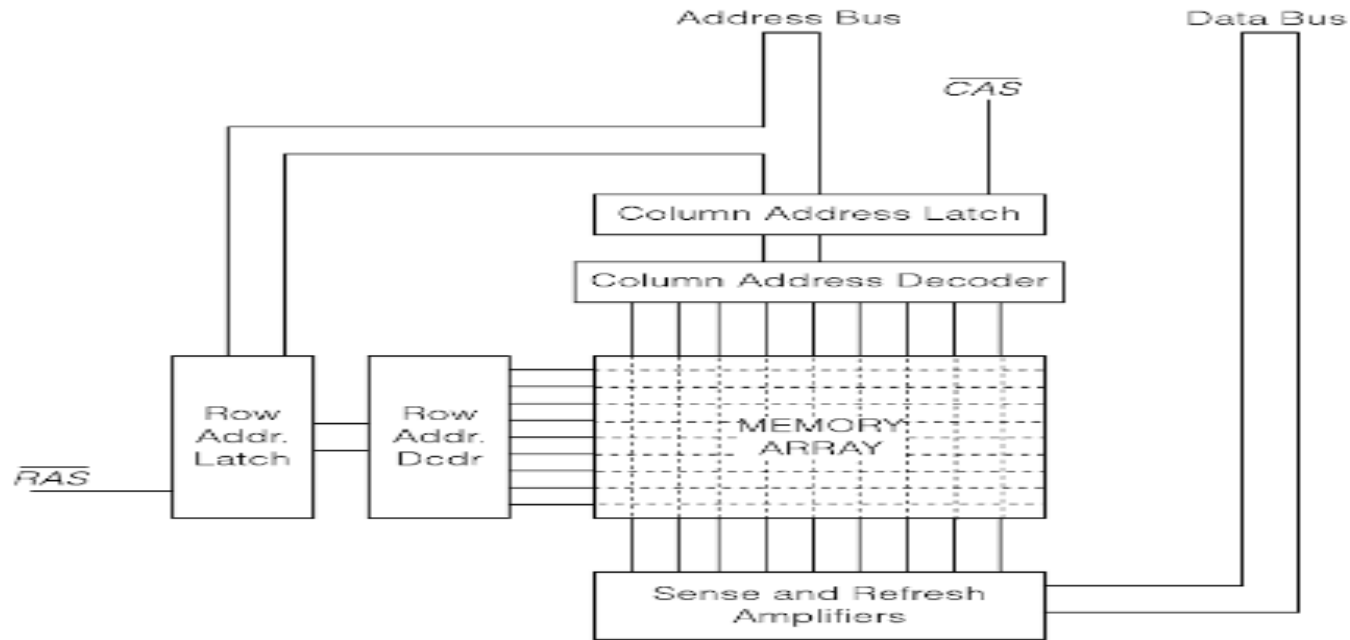
Memory Control of DRAM



Memory Control of DRAM

- Because row and column address information is placed on same address lines the pin count of DRAM is reduced.
- Each cell has a unique location or address defined by intersection of row and a column.

Internal Diagram DRAM Chip



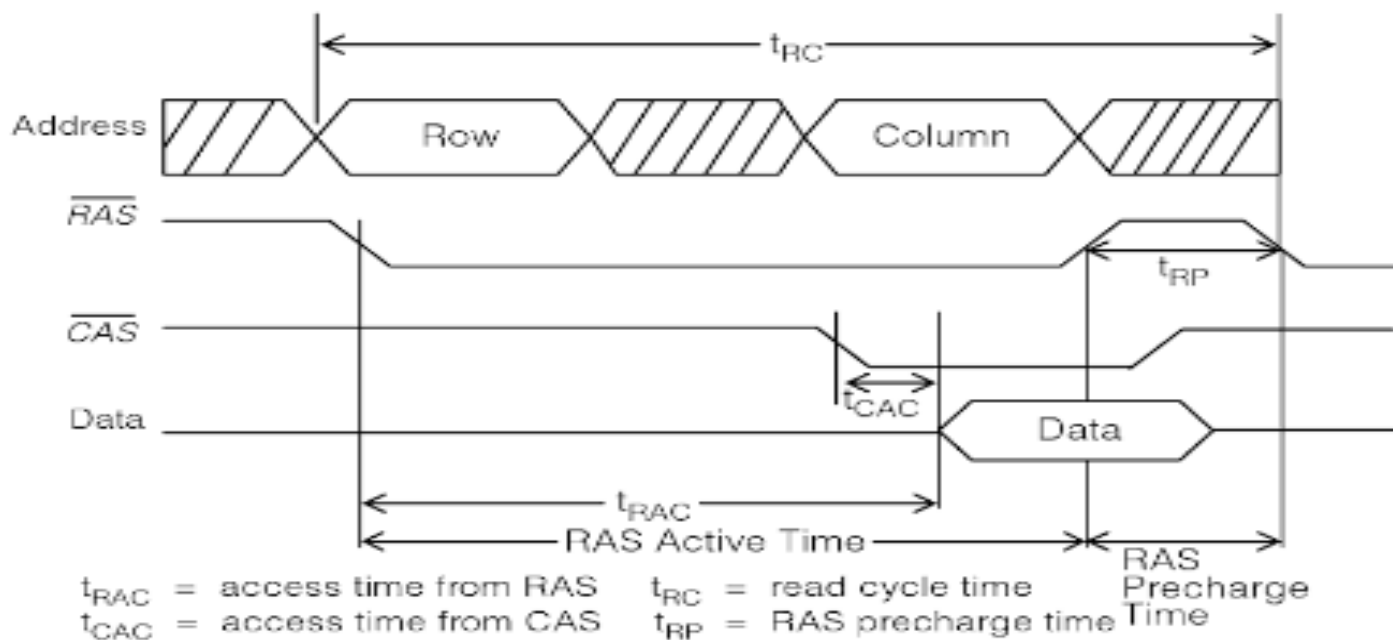
DRAM Read Cycle

- The row address is placed on address lines and given sufficient time to stabilize and be latched.
- The row address strobe $\overline{\text{RAS}}$ signal is activated.
- The row address decoder selects the proper row.
- The column address is placed on same address lines and allowed to stabilize and latched.

DRAM Read Cycle

- The $\overline{\text{CAS}}$ also serve as output enable, so once $\overline{\text{CAS}}$ signal has been stabilized, the sense amps place the data from selected row and column, on the data bus.
- The data in the selected address is available at output buffers of the chip and it is transferred to data bus.
- Before read cycle can be considered complete CAS and RAS must be returned to their previous state.

DRAM Read Cycle



DRAM Read Cycle

- T_{rac} = Access time for RAS

is time from the time RAS signal is activated to the time data is available on the data bus.

The read cycle time (T_{rc}) is sum of 'RAS active time' and 'RAS precharge time'.

RAS active Time- Time for which RAS signal is active.

DRAM Read Cycle

- RAS Precharge Time(t_{rp}):it is additional time needed before a new read cycle can be started.
- This is because there is a parasite capacitance for cell which must be precharged high before any operation to be commenced.
- The access time is referred to as latency.
- Packing Density is high as one transistor for one bit, while SRAM six T for one bit.

Refreshing

- Refreshing : Each row should be refreshed every 64msec

How is refreshing done?

There are many methods for refresh and one commonly used method is ROR (RAS only Refresh).

By activating each row using RAS.

DRAM Read Cycle

- DRAM controller takes care of scheduling the refreshes and making sure that they do not interfere with regular reads and writes
- To keep data in DRAM chip away, the DRAM controller periodically sweeps through all of rows by cycling repeatedly and placing a series of row addresses on address bus. This method is called ROR.
- To reduce the number of refresh cycles, one method of design is to split the address such that there are fewer rows and more columns.
- DRAM array is then a rectangular array rather than a square array.

Synchronous DRAM

- In asynchronous DRAM, access timing is not related to the system clock at all.
- In synchronous DRAM, access are synchronized with system clock and SDRAM is currently the RAM that is used as primary memory in general purpose computer systems.
- Embedded Systems do not have SDRAM inside them. But Many embedded boards do have SDRAM as external memory.

Synchronous DRAM

SDRAM has incorporated new features and modes of operation.

Synchronous Operation: all the operations are synchronized to the leading edge of system clock And thus controls are made easier.

Mode Register: There is a command register for this RAM into which common words are written to specify various operating modes and also generate various control signals. It makes RAM 'programmable'.

Synchronous Vs Asynchronous DRAM

Asynchronous

- Does not share any common clock with CPU, the controller chips have to manipulate the DRAM's control pins based on all sorts of timing considerations.
- For Accessing Memory, toggling of the external control inputs has a direct effect on internal memory array.

synchronous

- Shares a common clock with CPU, commands can be placed on its control pins on clock edges.
- In SDRAM, the input signals are latched into control logic block which functions as input to a state machine.
- State Machine controls memory Access.
- Read, write and refresh are initiated by loading control commands into device.

DDR (Double Data Rate) SDRAM

- It can be made to transfer data at rising and falling edges of the clock, instead of just at rising edge.
- This should double the data rate.
- It achieves higher throughput by using differential pairs of signal wires that allows faster signaling without noise and interference problems.
- **DDR-2 and DDR-3:** Just double data rate as with DDR. The additional signals required for differential pairs add to pin count of DDR2 and DDR3.

ROM(Read Only Memory)

- User burn the contents of this which is not lost when power is switched off.
- Firmware is term for ROM and its content together.
- Type of **programmable memory**. It has internal fuses which can be blown , create a bit pattern which is permanent and hence can be read whenever needed.
- If it is OTP, its contents can never be changed again.

EPROM(Erasable and programmable)

Contents can be erased by exposing to ultraviolet radiation. Such ROMs have a window through which UV light penetrates the chip.

EEPROM (Electrically Erasable PROM)

- Erasure can be done while chip is on circuit board.
- Programmer can change the data one byte at a time, takes long time when erasing.
- EEPROM is non-volatile, but also erasable and reprogrammable
- Used for data storage in small quantities where data have to be read frequently, may not have to be changed normally.
- Dedicated EEPROMs chips are available which can be connected to MCUs
- Two types:
 - Parallel:** data lines are connected directly to processor's data lines for data transfer
 - Serial:** One/Two data lines

PIC MCUs with EEPROMs inside the chip

Not very large but to store data like constants, identification numbers etc.

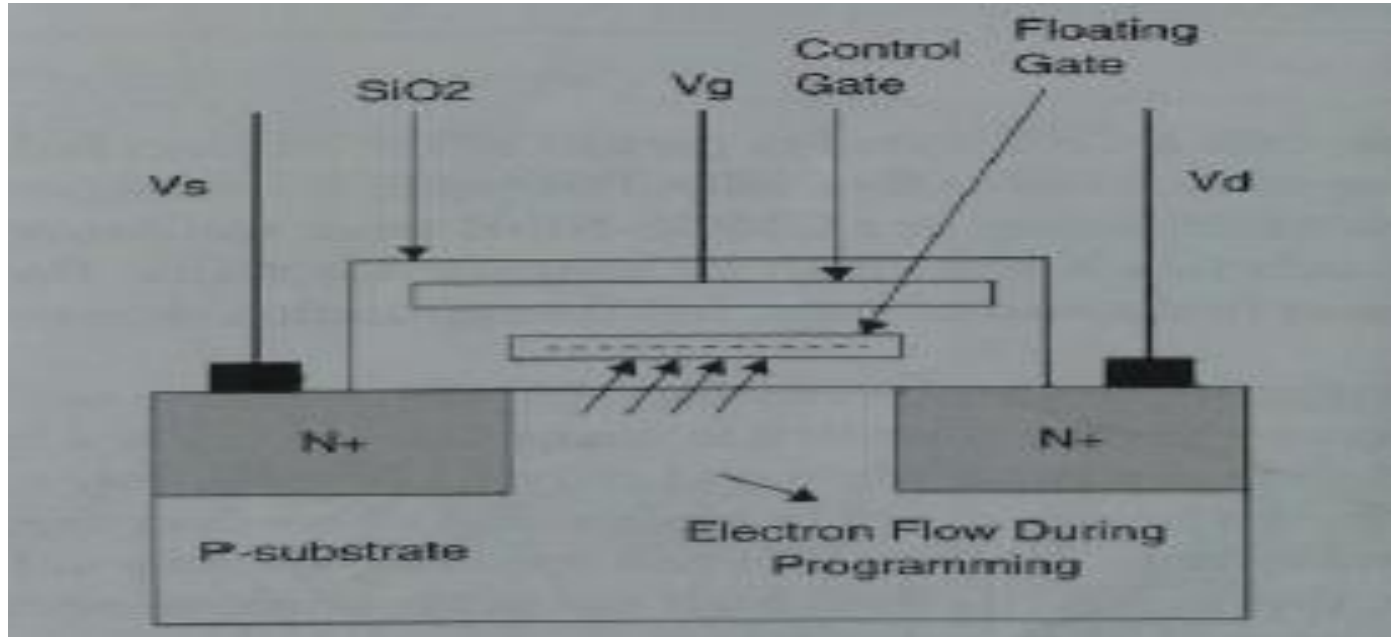
Flash Memory

- Non-volatile memory.
- **Flash memory** is a solid-state chip that maintains stored data without any external power source. It is commonly used in portable electronics and removable storage devices.
- Similar to EEPROM, but difference is EEPROM can be erased and written only one byte at a time, while for flash these operations are 'block oriented'.

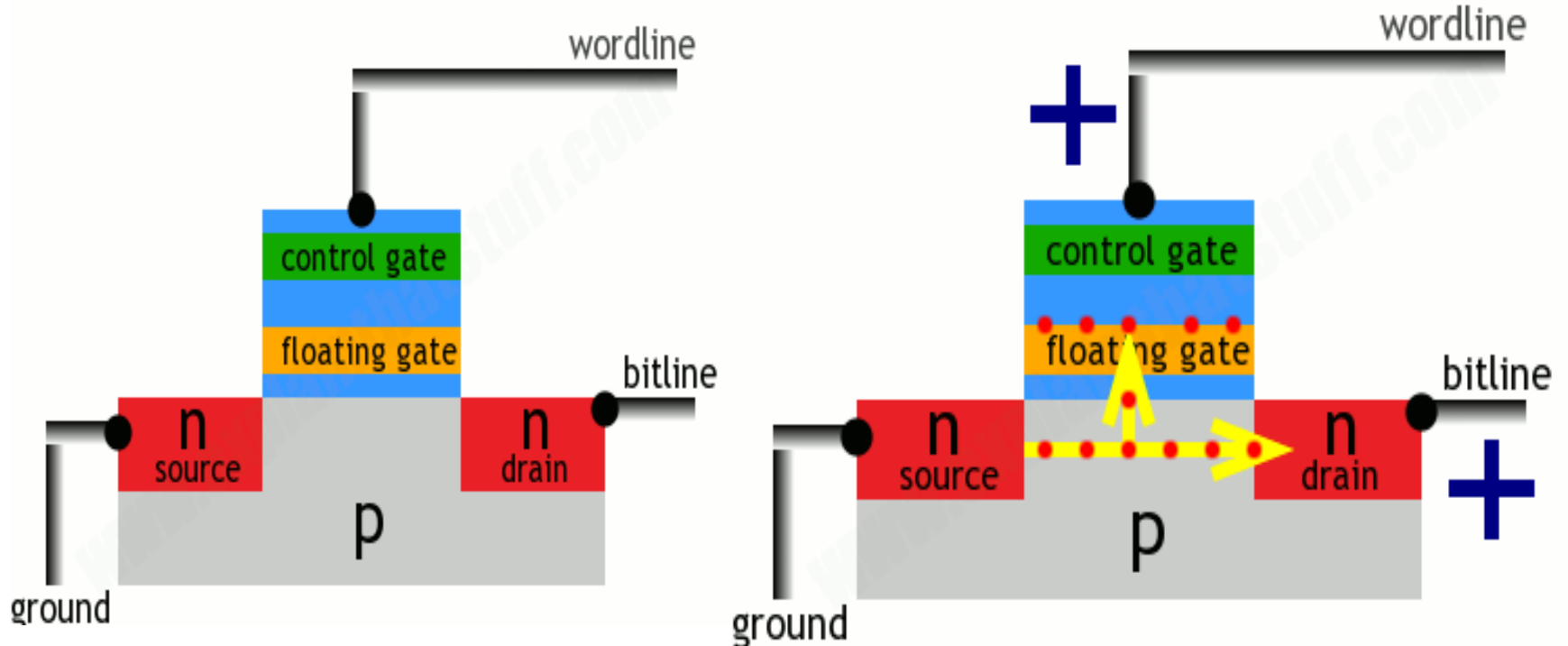
Flash Memory Cell

- Flash memory stores data in an array of memory cells.
- Memory cells are made from floating gate MOSFETs(or FG MOS)
- Floating gate MOSFETs can store electrical charge for years, without the need for a power supply.

Flash Memory Cell



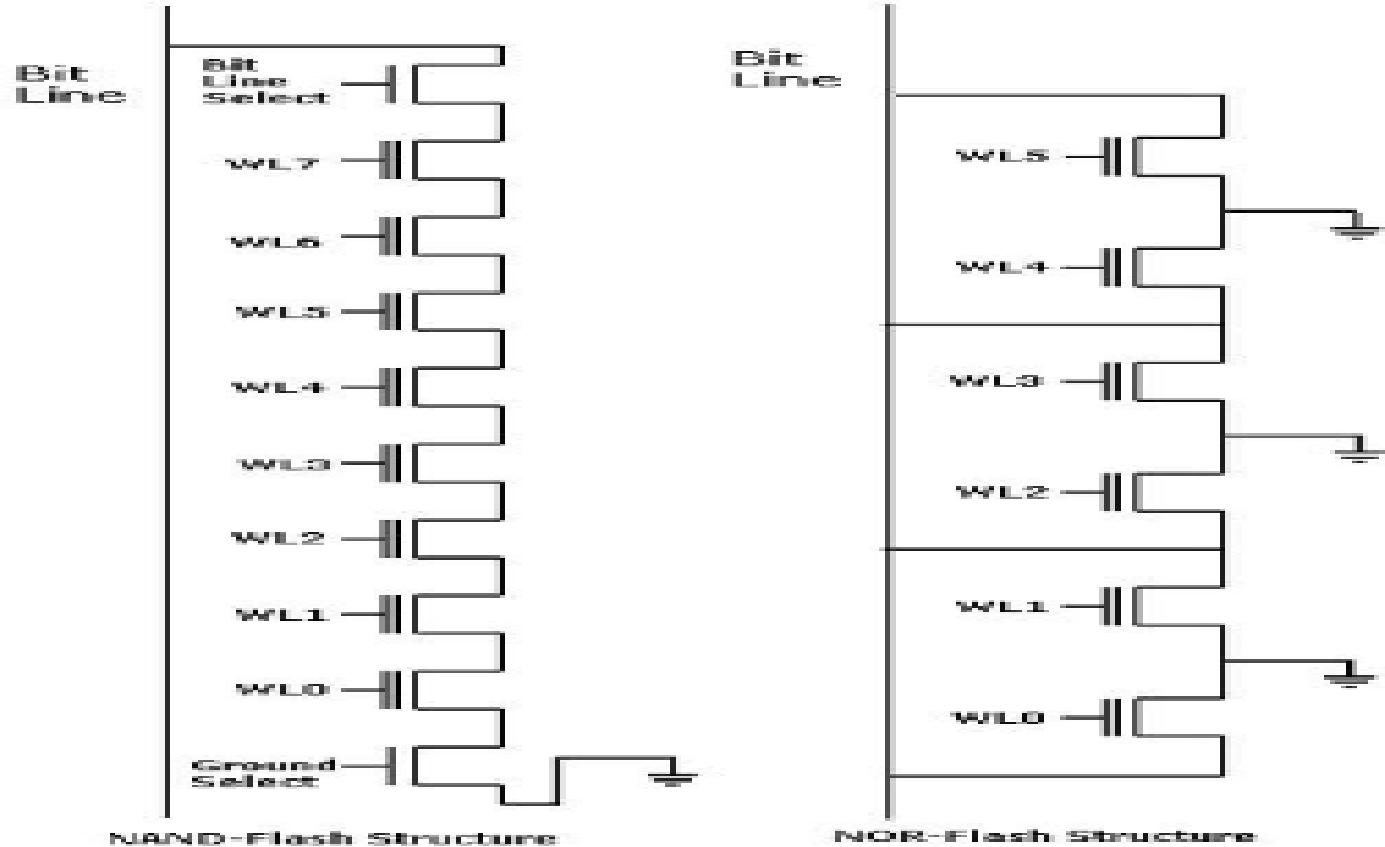
Flash memory working



Principle of Data Storage in Flash Memory

- Both the source and the drain regions are rich in electrons (because they're made of n-type silicon), but electrons cannot flow from source to drain because of the electron deficient, p-type material between them.
- If we apply a positive voltage to the transistor's two contacts, called the **bitline** and the **wordline**, electrons get pulled in a rush from source to drain. A few also manage to wriggle through the oxide layer by a process called tunneling and get stuck on the floating gate
- The electrons will stay on the floating gate indefinitely, even when the positive voltages are removed and whether there is power supplied to the circuit or not. If we disconnect the positive voltages from the bitline and wordline and try to pass a current through the transistor, from source to drain, none will flow. the electrons on the floating gate will stop it. So, in this state, we say the transistor is storing a zero.
- The electrons on the floating gate can be flushed out by putting a negative voltage on the wordline. This repels the electrons back the way they came, clearing the floating gate and allowing a current to flow through the transistor once again. In this state, we say the transistor is storing a one.

NOR and NAND Flash Structure



NAND and NOR flash Memory

NAND Flash

- Originated from Toshiba Technology.
- Cells are connected in series
- Sequential access
- Read or written only as blocks
- Used for : USB, memory cards, solid state drives, mobile phone , Digital Camera

NOR Flash

- Developed by Intel
- Cells are connected in parallel to bit line facilitating reading/ writing/ erasing of each cell individually.
- NOR is higher speed and random access is possible
- data can be read and written in quantities **as small as byte.**
- Used for: TO store BIOS in PCs and operating System in PDAs and Mobile phones, embedded systems

SD (Secure Digital) Cards

- Flash memory with security features incorporated
- Addition of cryptographic security protection for copyrighted data/music
- SD card Association sets specification for SD cards
- Available in various sizes and capacities
- Designed as mini SD, micro SD, MMC (multi media card) , mobile MMC etc.

Comparing Volatile and Non-volatile Memory

- MCUs have a small amount of SRAM and large amount of flash.
- SRAM is used for small amount of storage of intermediate data during computation
- Flash memory is area where code is burned, part of it used for data storage
- Small amount of EEPROM used to hold data which is unlikely to change.
- In 8051
 - Maximum size of SRAM- 256 bytes
 - Flash ROM- 64 KB
 - And no EEPROM

Comparison of Different Type of Memories

Memory Type	Features	Usage
SRAM	Volatile, high speed	Cache On chip RAM in MCUs
DRAM	Volatile, speed lower than SRAM	Primary memory in PCs External memory in embedded boards
EEPROM	Non-volatile, very low speed	Storage of small amounts of data which remains more or less constant
FLASH ROM	Non-volatile, lower speed than RAMs	i) Storage of large amounts of data which may have to be changed frequently ii) Storage of the program code of embedded systems

Memory Shadowing

- In PCs BIOS is in flash, which is a relatively slow memory device.
- Device drivers which are part of BIOS are frequently used, so having to access ROM each time creates a quite delay.
- To circumvent, BIOS is copied to RAM and ROM access are disabled.
- RAM used for this is called shadow RAM and this technique is called memory shadowing.

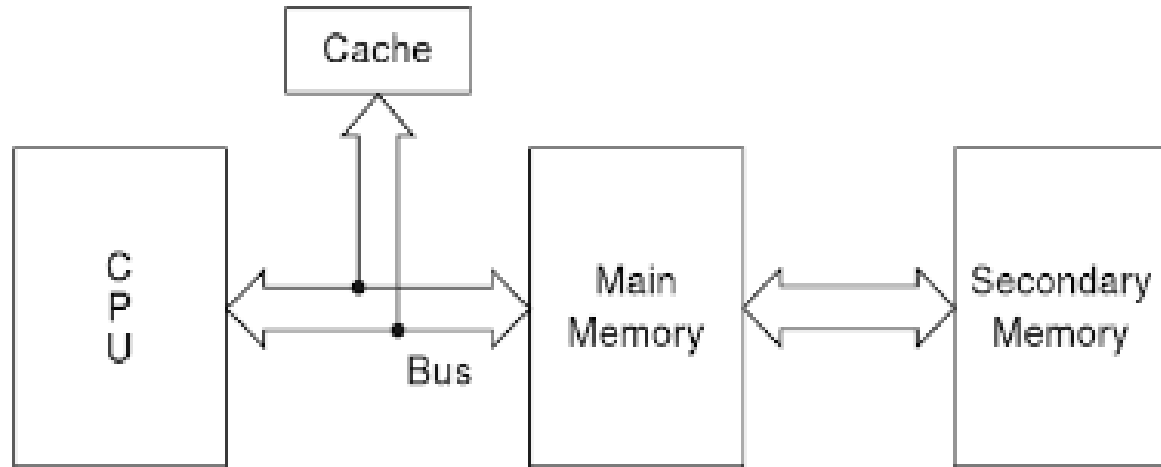
Cache

- Ideal condition is to find data in cache called hit
- Certain times required information not found in cache called miss then taken from main memory and relatively a slow access.
- **Cache mapping policy** is designed to maximize hits— as cache size increases or mapping policy becomes more stringent, the amount of hardware required for managing this increases.

Cache

- Thus, the amount of cache required for a system and amount it can afford (in terms of increased hardware budget) have to be carefully balanced.
- Cache memory is closet to processor and also fastest.
- But content of cache is only copy of portion of main memory.
- Cache memory uses SRAM
- Main memory -DRAM
- Different levels of caches –L1, L2, and L3 with L1 being closet and L3 being farthest from CPU.

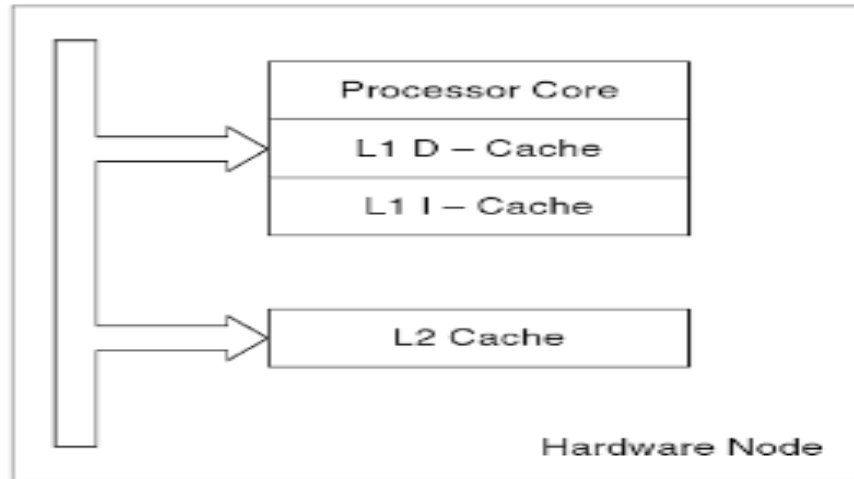
Memory Hierarchy in System With Cache



Embedded Systems And Cache

- DO Embedded Systems Have Cache?
- Answer is only mid and high range system posses it or need it
- Many high end processors specially DSP processor have it in their cores.

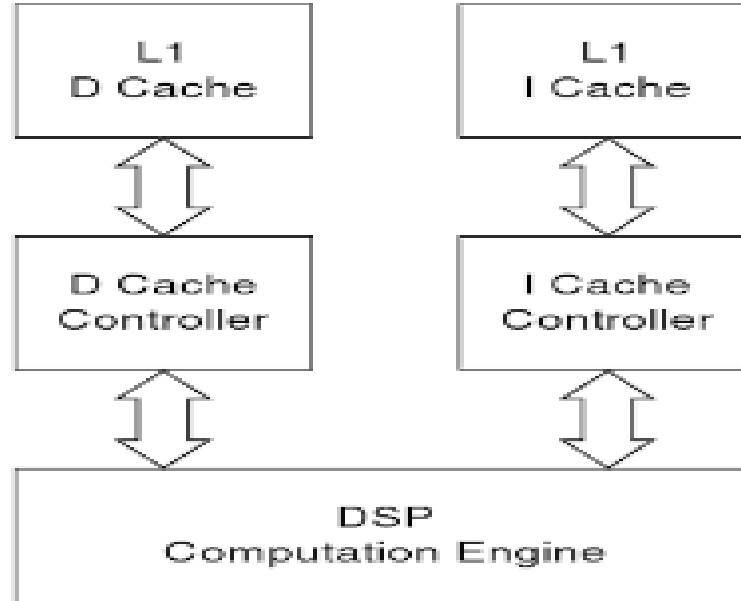
A hardware node of Embedded System



A hardware node of Embedded System

- Fig shows embedded processor board or a node in a networked systems.
- There is a processor in the system and the processor has an on chip L1 cache which is divided into **an instruction cache and data cache**
- **L2 cache is outside the processor .**
- DSP processors are one class of processors which mandatorily use cache internally.
- Internals of DSP processor with L1 caches and their respective controller are shown in next slide.

DSP processor with Cache



Caches and Real-time Systems

- Caches are not a preferred items for real time systems because of the probabilistic nature of operations of caches.
- Every thing is fine in case of hit –but in case of a miss, access becomes very slow.
- Data in cache is continually changed, there is no way of guaranteeing that a particular item will be there when needed. Real-time systems are time critical and are to be designed to be deterministic.
- Processor used for real time applications usually have a ‘tightly coupled memory’ which gives deterministic performance in contrast to the probabilistic performance of caches.

Tightly Coupled Memory

- A fast memory directly connected to the processor core (**inside the chip**)
- It is meant to provide low latency memory that the processor can use without the unpredictability associated with caches. Actually it is memory in which critical items like interrupt handlers, real-time task codes etc are stored.
- It may also contain the data which is **frequently needed**
- **ARM processors are designed to be specific for real-time applications have TCM and associated registers.**

Low Power Design

- Every device used in the system must be able to work with lowest power supply voltage and lowest current. This applies to processors and peripherals
- MCU with low power processor core
 - 32-bit ARM core
 - 16- bit TI's MSP 430

Factors which tend to increase power dissipation and attempt to reduce

i. **High Frequency:**

- a. Use the minimum clock rate which is actually needed.
- b. Some applications don't need the same high clock rates as certain others.
- c. Trick is to use a processor whose clock frequency can be dynamically changed.
- d. Modern day processors have internal PLLs (phase Locked loops) that can change the clock frequency on the fly.

Factors which tend to increase power dissipation and attempt to reduce

ii. High power supply voltage:

High currents and so more power.

Low power supply voltages in range of 3V . This reduces noise margins.

We can use differential signaling to improve noise resilience

iii. Complex Hardware:

- a. Simpler hardware directly translates to low power there are less number of transistors and so less is power dissipated.
- b. RISC cores are comparatively low power cores

Factors which tend to increase power dissipation and attempt to reduce

iv. Higher Bus Widths:

More bus lines means more capacitances getting charged and discharged for change of logic states, it's best to reduce bus widths of external buses.

v. I/O devices:

Try to avoid I/O devices which consumes more power (leakage and quiescent current ratings should be verified and which offer lowest must be chosen)

e.g. I/O devices such as optical isolators, electromechanical relays, back lit display etc

Factors which tend to increase power dissipation and attempt to reduce

vi. Circuit Design:

As processor are designed to switch off the circuitry which is not being actively used. **The same idea should be used in circuits in the system design.**

Using gated clock help i.e. when clock is removed from part of circuit it does not function thus needs no power

Factors which tend to increase power dissipation and attempt to reduce

vi. Using low power modes of processor:

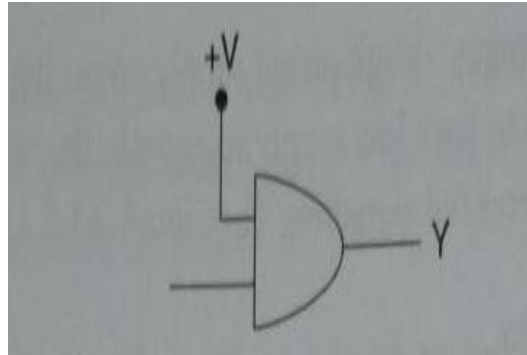
- a. systems don't need to operate continuously. Many wake up on interrupt .
- b. Sleep and power down modes of processor must be used effectively.

vii. Battery Type:

In some cases recharging is possible in others , it is not possible.

Both need different types of batteries.

Floating State Of input

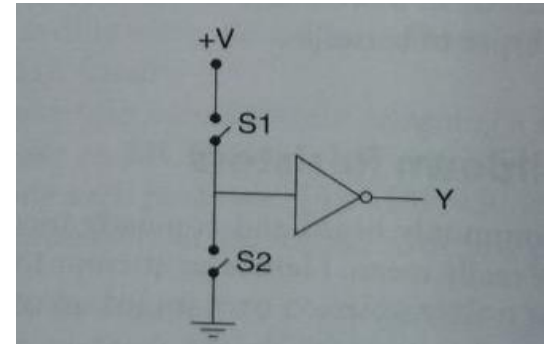


Floating State Of input

- One input is connected to supply and other left open
- Assumption of leaving pin open is equivalent to connected to 0 is wrong
- But a floating input really floats and cannot take fixed value permanently and usually found to have high level voltage
- Susceptible to accepting noise voltage pulse and keep changing

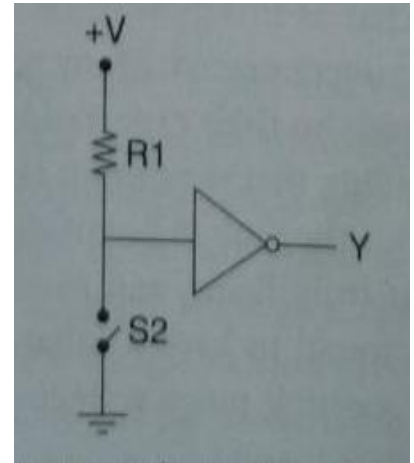
Pulldown and Pullup Resistance

- Connected to 1 or 0 depending on which switch is closed.
- In some cases there may be reason to limit the current that flows to the input so need of pull up or pull down resistor.



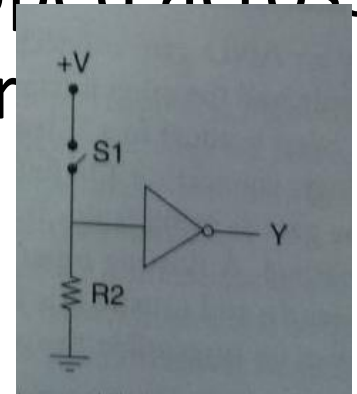
Pull up resistor

- Here input has been pulled up to high through resistor
- When input is to be high s2 is kept open



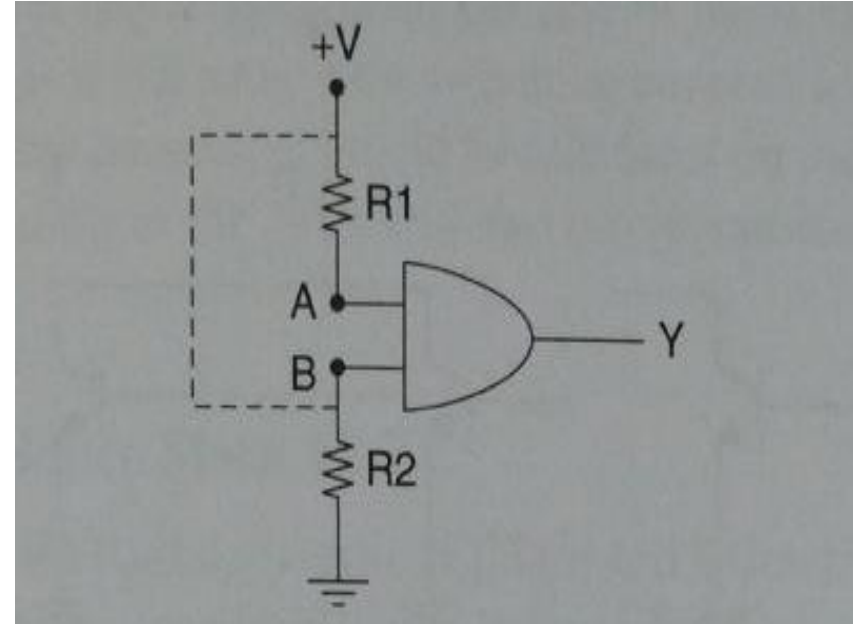
Pulldown Resistor

- R2 is pull down resistor as it is connected to ground
- When s1 is open input is truly grounded
- When s1 is closed, voltage v is dropped across R2, corresponding to 1 input but current limited by R2



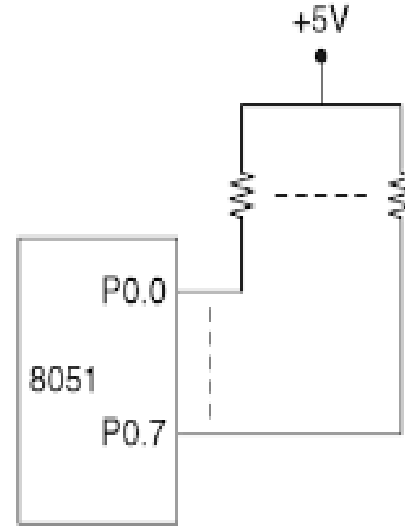
AND gate with Pullup and Pulldown resistors

- If we ignore dotted line
- A and B are connected to 1 and 0 respectively
- Connect B to 1 by V volt also available at B but current is limited by resistance R2.



Need for such circuits

- In 8051 port 0 needs pull up resistances as drains of output transistors are left open.
- Separate pullup resistors are used , as port pins may have to be used as single bit port lines.

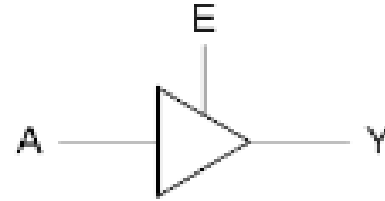


Weak and Strong Pullup

- When current drawn through pull up is small, it is weak pullup otherwise strong pull up
- Weak pull up i.e. high resistance
- A weak pull up makes switching speed slow so not advisable as it will slow the switching between levels.
- Values of pullup resistance must be calculated on basis of speed requirements, current specification of gates and no. of o/ps connected together

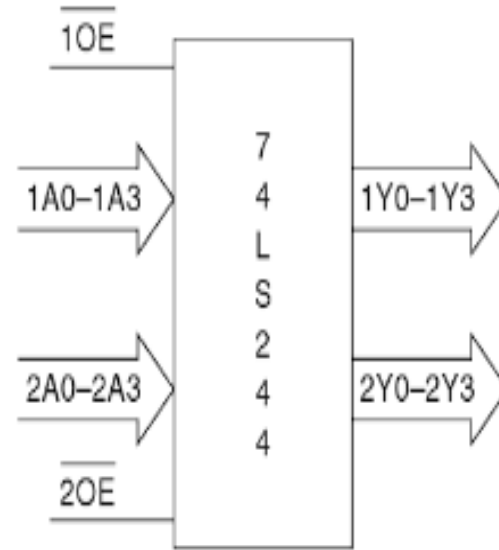
High impedance state, Hi-Z

- Tri state buffer
- If it is put in enabling state the signal at A is transferred to Y
- Other wise the buffer act like open circuit



High impedance state, Hi-Z

- There is enable pin OE when OE is low, output pins are active and take on logical values of input
- When OE is high, output pins are at high impedance state
- The device connected to these pins is disconnected and interconnection lines are actually open circuit



High impedance state, Hi-Z

- Only one of the devices should be connected to bus
- And this is achieved by selectively enabling each device.
- When enabling pin of device is inactive the output lines are in high impedance state and no connection exists between data bus and device output.
- this ensures only one device can send data to MPU at a time

