INTRODUCTION TO MICROCONTROLLERS

COURSE OBJECTIVES

- DESCRIBE THE ARCHITECTURE AND PROGRAMMING MODEL OF 8051 AND ARM PROCESSOR
- DESCRIBE THE HARDWARE DETAILS OF 8051 AND ARM PROCESSOR
- WRITE ASSEMBLY LANGUAGE PROGRAMS FOR 8051 AND ARM PROCESSOR
- WRITE PROGRAMS FOR INTERFACING EXTERNAL I/O DEVICES TO 8051 AND ARM PROCESSOR
- DISCUSS THE WORKING PRINCIPLES AND APPLICATION OF ON-CHIP PERIPHERALS OF 8051 AND ARM PROCESSOR

EXPECTATIONS FROM A MICROCONTROLLER

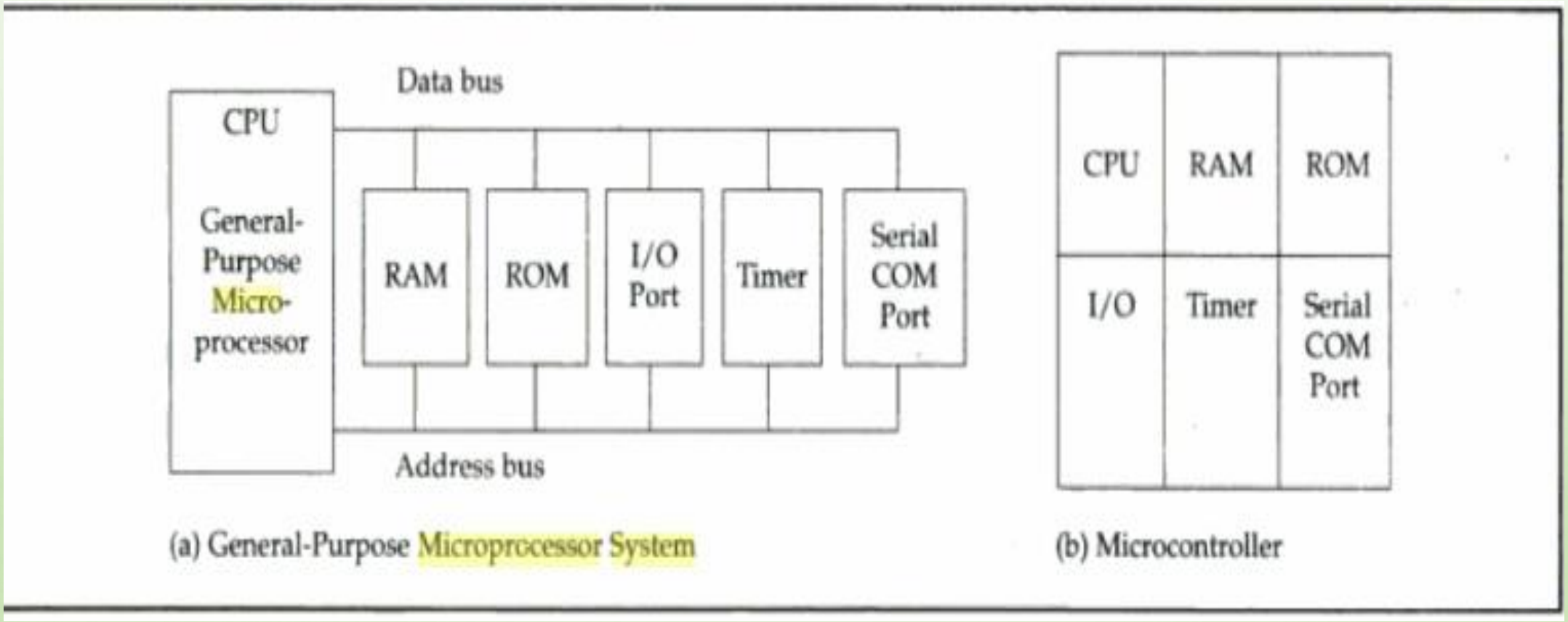
- LETS CONSIDER WHAT ARCHITECTURAL FEATURES WOULD BE NEEDED IN A MICROCONTROLLER.
- WHAT ARE THE EXPECTED APPLICATIONS?
  - SENSING THE ENVIRONMENT (INPUT)
  - PRODUCING A RESPONSE (OUTPUT)
  - DO AN ACTIVITY REPEATEDLY (TIMER/COUNTER)
  - PRIORITIZED RESPONSE (INTERRUPTS)
  - SOFTWARE TO CONTROL THE PROCESS (NON-VOLATILE MEMORY)
  - TEMPORARY DATA (RAM)

COMPARING PROCESSORS AND CONTROLLERS

Microprocessors

- processing unit (CPU) only
- Needs many ICs to implement a small system

COMPARING PROCESSORS AND CONTROLLERS



COMPARING PROCESSORS AND CONTROLLERS

- 16 BIT ALU
- FEW GENERAL PURPOSE REGISTERS AND A FEW SPECIFIC PURPOSE REGISTERS
- NO INTERNAL RAM OR ROM
- NO OTHER HARDWARE ON CHIP

COMPARING PROCESSORS AND CONTROLLERS

Microcontrollers

- CPU + I/O + Timer(s) [+ ROM] [+ RAM]
- Low to moderate performance only
- Limited RAM space, ROM space and I/O pins
- EPROM version available
- Low chip-count to implement a small system
- Low-cost at large quantities
- Development tools readily available at reasonable cost

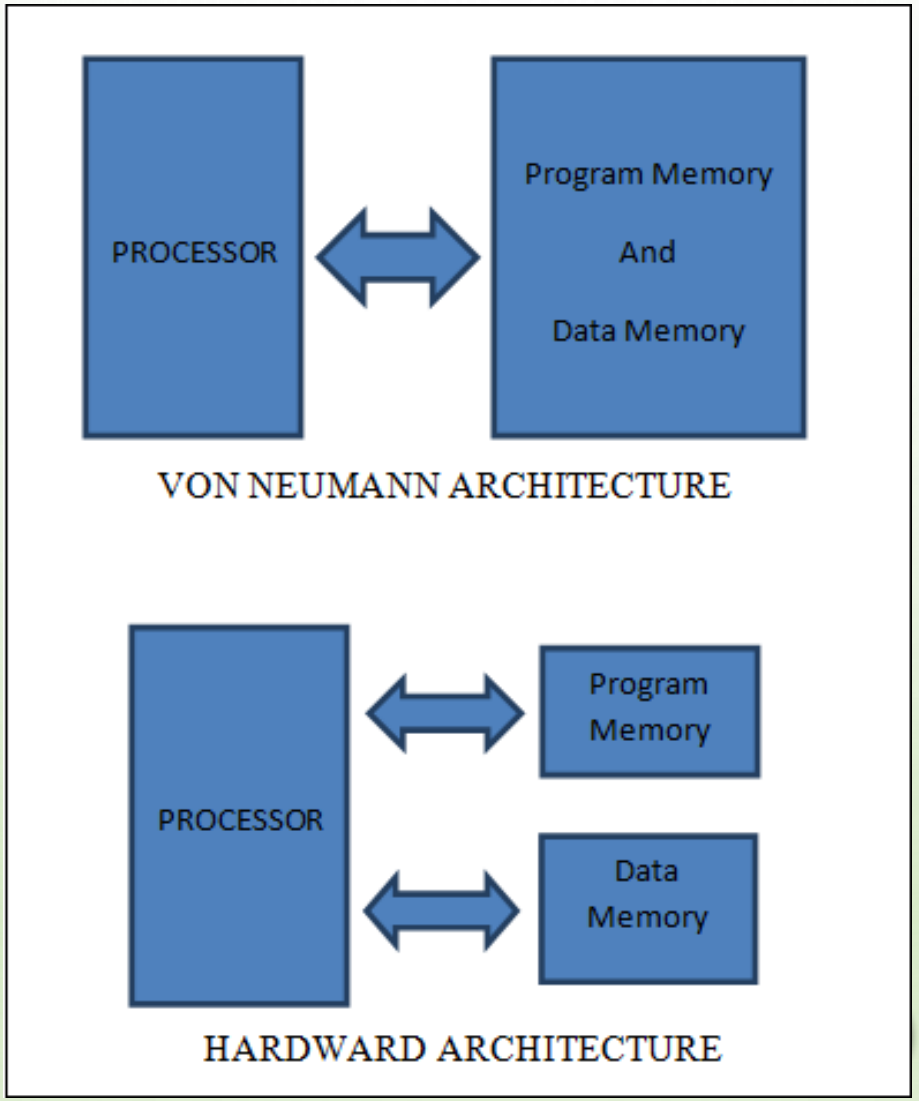
COMPARING PROCESSORS AND CONTROLLERS

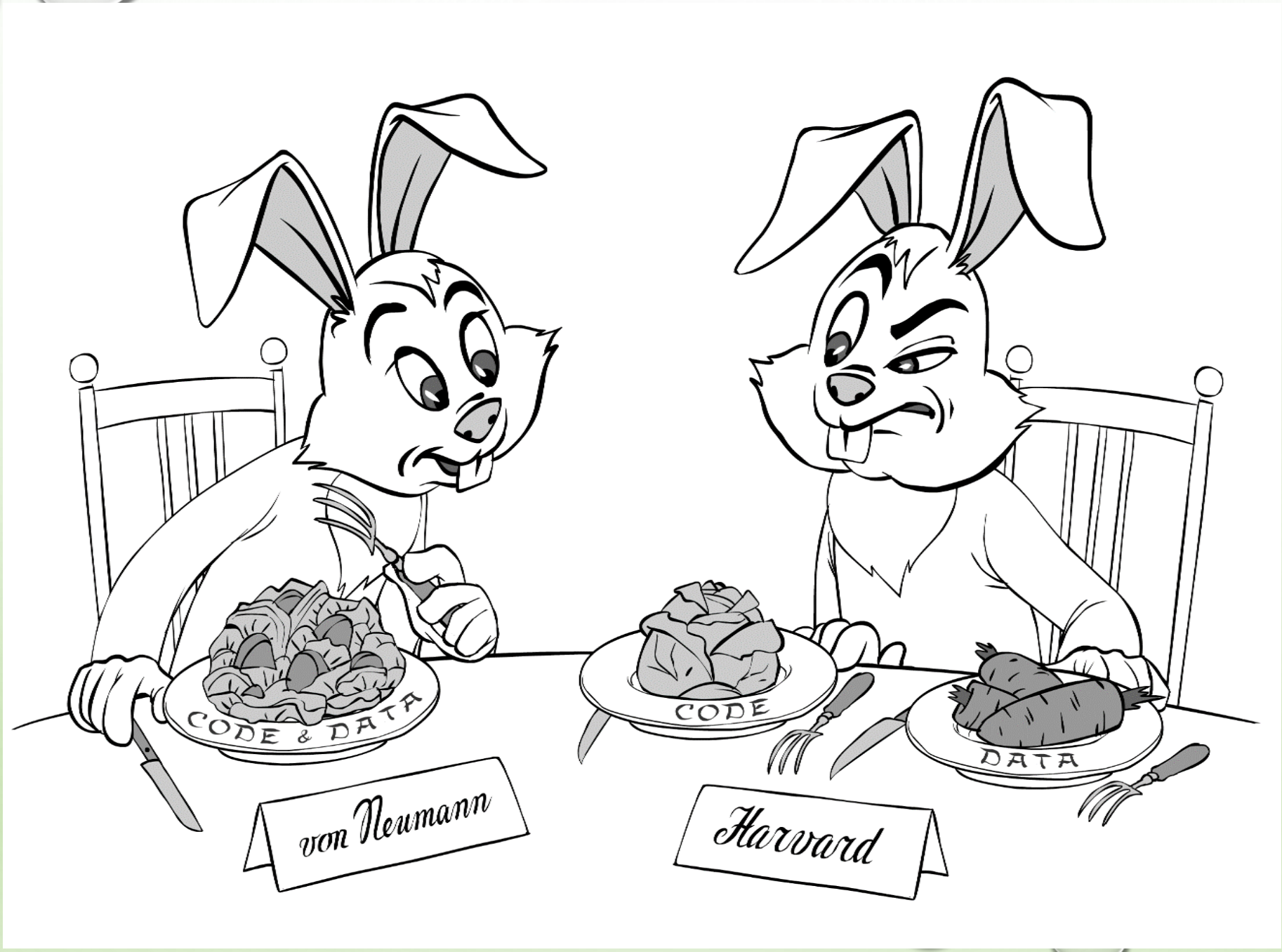
- 8-bit CPU optimized for control applications
- Capability for single bit Boolean operations.
- Supports up to 64K of program memory.
- Supports up to 64K of program memory.
- 4 K bytes of on-chip program memory.
  - Newer devices provide more.
- 128 or 256 bytes of on-chip data RAM
- Four 8 bit ports.
- Two 16-bit timer/counters
- UART
- Interrupts
- On-chip clock oscillator

Microprocessor	Microcontroller
A silicon chip representing a Central Processing Unit (CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions	A microcontroller is a highly integrated chip that contains a CPU, scratch pad RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports
It is a dependent unit. It requires the combination of other chips like Timers, Program and data memory chips, Interrupt controllers etc for functioning	It is a self contained unit and it doesn't require external Interrupt Controller, Timer, UART etc for its functioning
Most of the time general purpose in design and operation	Mostly application oriented or domain specific
Doesn't contain a built in I/O port. The I/O Port functionality needs to be implemented with the help of external Programmable Peripheral Interface Chips like 8255	Most of the processors contain multiple built-in I/O ports which can be operated as a single 8 or 16 or 32 bit Port or as individual port pins
Targeted for high end market where performance is important	Targeted for embedded market where performance is not so critical (At present this demarcation is invalid)
Limited power saving options compared to microcontrollers	Includes lot of power saving features

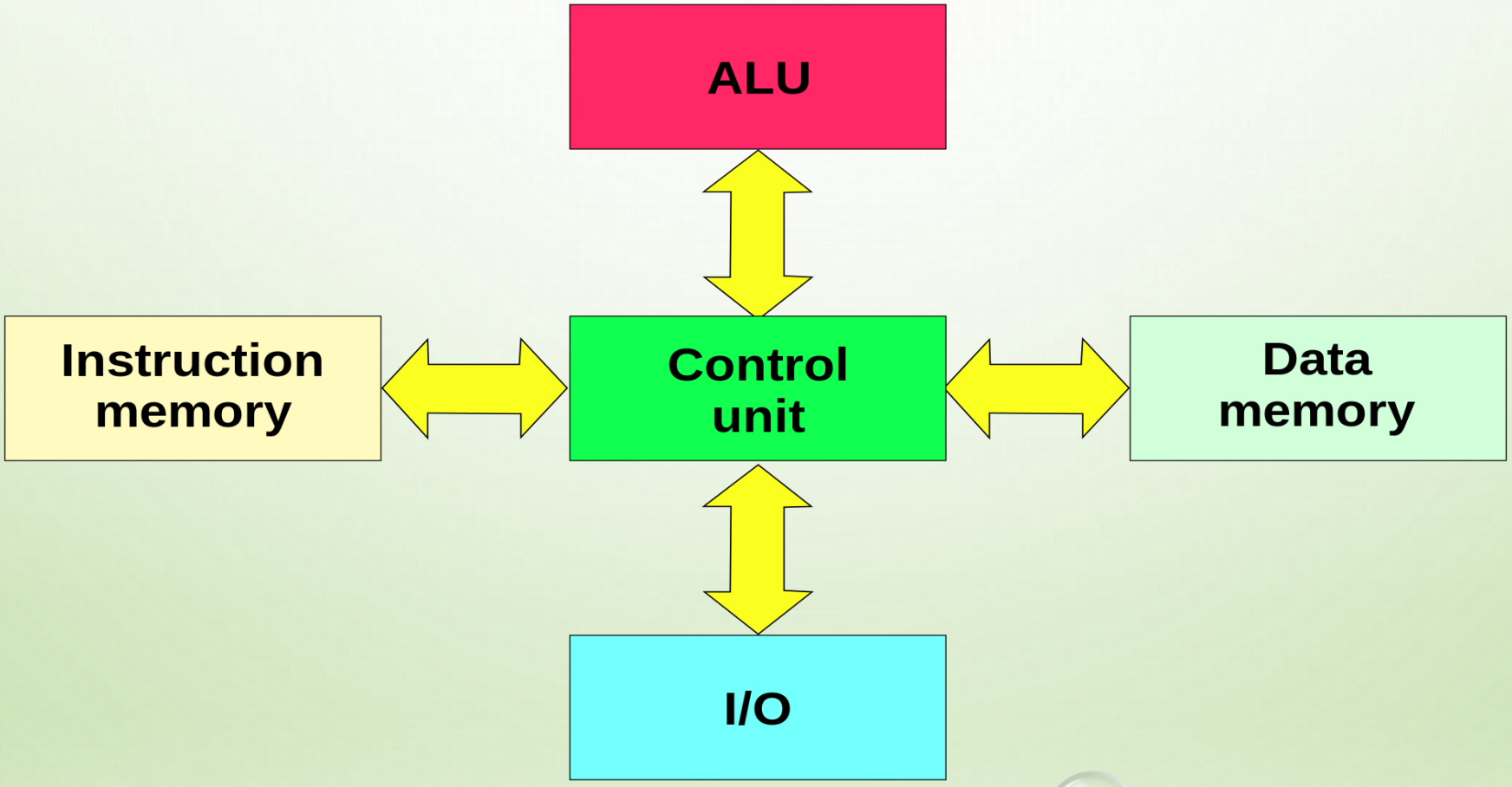
RISC	CISC
Lesser no. of instructions	Greater no. of Instructions
Instruction Pipelining and increased execution speed	Generally no instruction pipelining feature
Orthogonal Instruction Set (Allows each instruction to operate on any register and use any addressing mode)	Non Orthogonal Instruction Set (All instructions are not allowed to operate on any register and use any addressing mode. It is instruction specific)
Operations are performed on registers only, the only memory operations are load and store	Operations are performed on registers or memory depending on the instruction
Large number of registers are available	Limited no. of general purpose registers
Programmer needs to write more code to execute a task since the instructions are simpler ones	Instructions are like macros in C language. A programmer can achieve the desired functionality with a single instruction which in turn provides the effect of using more simpler single instructions in RISC
Single, Fixed length Instructions	Variable length Instructions
Less Silicon usage and pin count	More silicon usage since more additional decoder logic is required to implement the complex instruction decoding.
With Harvard Architecture	Can be Harvard or Von-Neumann Architecture

VON NEUMANN AND HARVARD ARCHITECTURE




CODE & DATA
CODE
DATA
von Neumann
Harvard

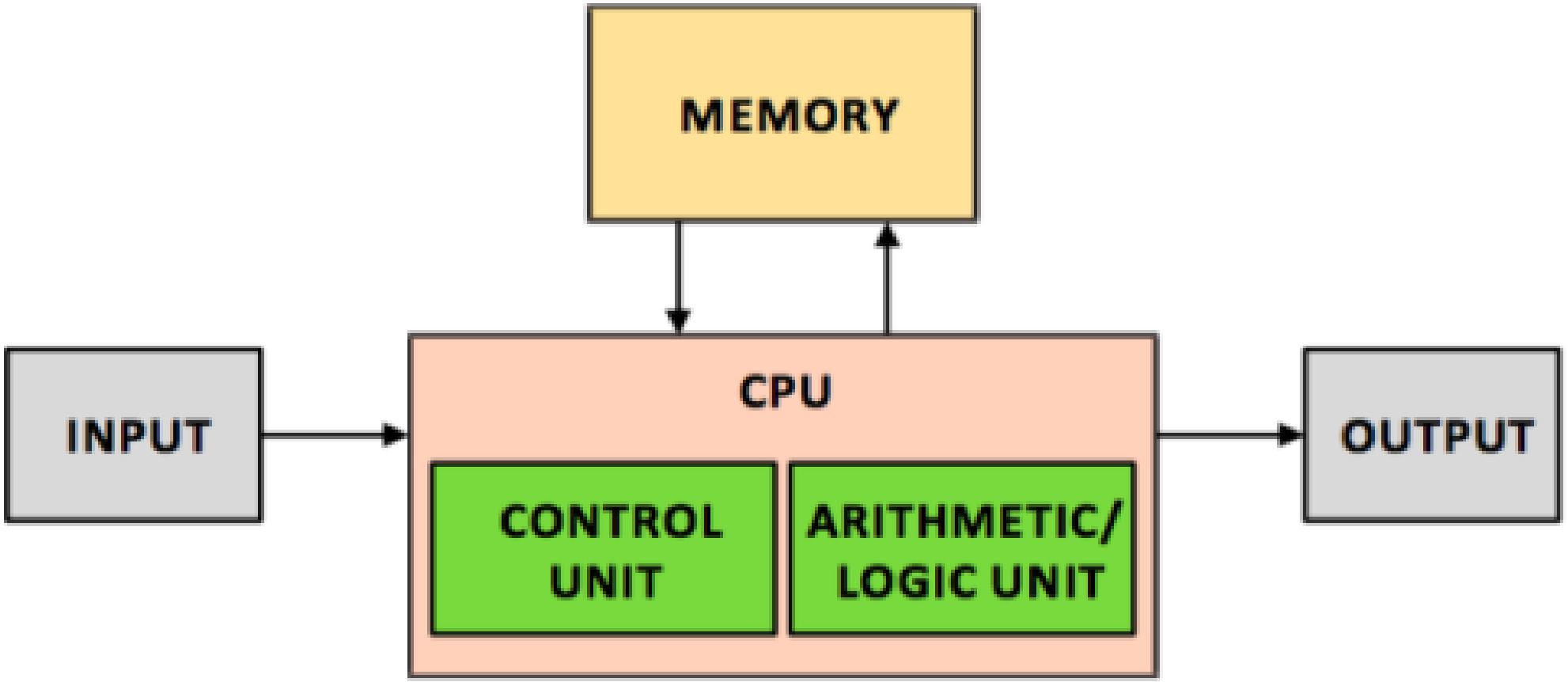
HARVARD ARCHITECTURE



HARVARD ARCHITECTURE

- HARVARD CAN'T USE SELF MODIFYING CODE
- ALLOWS TWO SIMULTANEOUS MEMORY FETCHES
- MOST DSP'S USE HARVARD FOR STREAMING DATA
  - GREATER MEMORY BANDWIDTH
  - DIFFERENT MEMORY BIT DEPTH BETWEEN INSTRUCTIONS AND DATA
  - MORE PREDICTABLE BANDWIDTH
- EASY TO IMPLEMENT PIPELINING
- CAN HAVE AN EFFICIENT PIPELINING

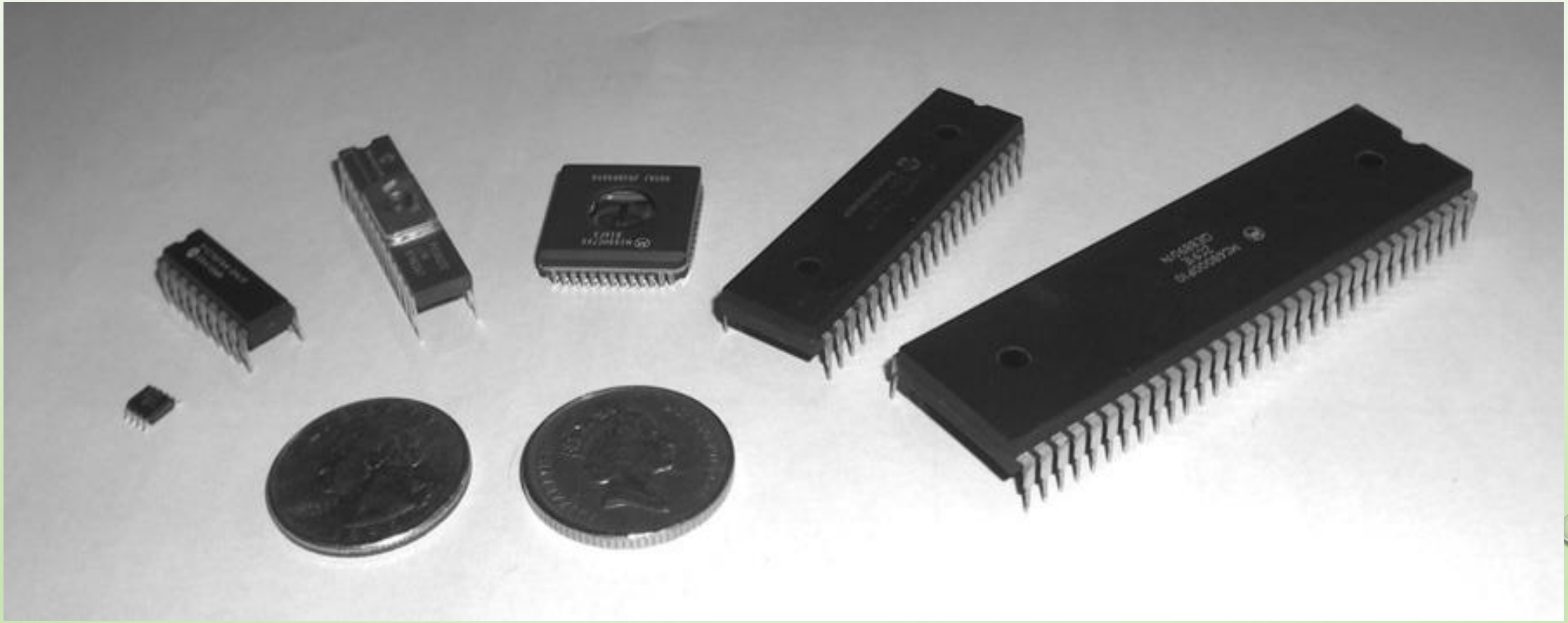
VON NEUMANN ARCHITECTURE



VON NEUMANN ARCHITECTURE

- THEY HAVE SHARED SIGNALS AND MEMORY, FOR CODE AND DATA
- PROGRAM CAN BE EASILY MODIFIED BY ITSELF SINCE IT IS STORED IN READ-WRITE MEMORY
- MORE FLEXIBLE AND EASY TO IMPLEMENT
- SUITABLE FOR GENERAL PURPOSE PROCESSORS

PROCESSORS PACKAGING

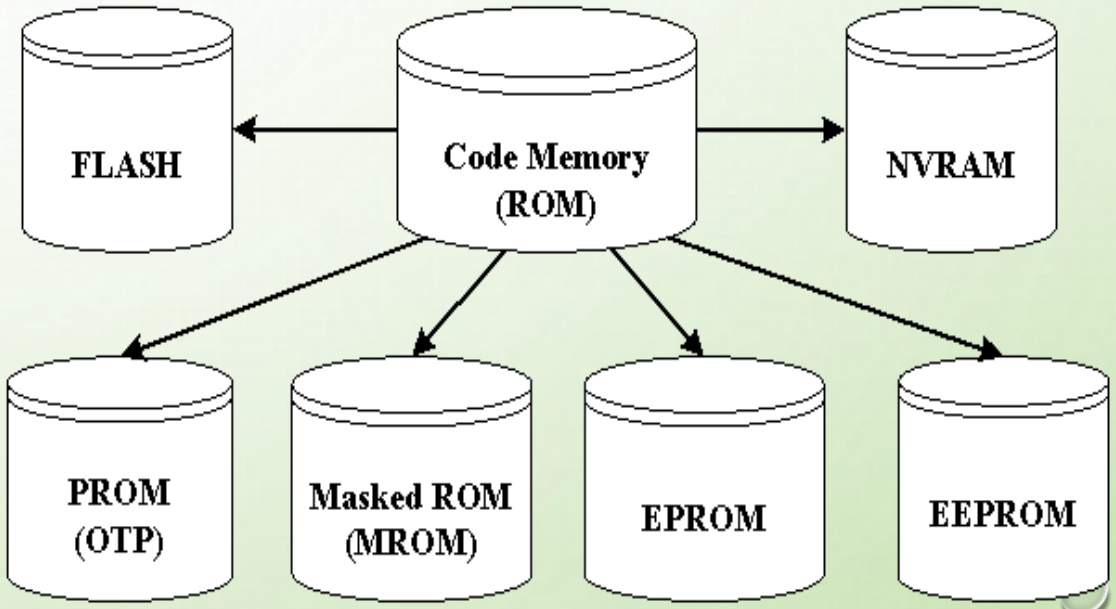


MEMORY – OFF CHIP, ON CHIP, RAM AND ROM

- MEMORY IS AN IMPORTANT PART OF AN EMBEDDED SYSTEM.
- CERTAIN EMBEDDED PROCESSORS/CONTROLLERS CONTAIN BUILT IN PROGRAM MEMORY AND DATA MEMORY AND THIS MEMORY IS KNOWN AS ON-CHIP MEMORY.
- OTHERS DO NOT CONTAIN ANY MEMORY INSIDE THE CHIP AND REQUIRES EXTERNAL MEMORY TO BE CONNECTED WITH THE CONTROLLER/ PROCESSOR TO STORE THE CONTROL ALGORITHM. IT IS CALLED OFF CHIP MEMORY.
- THE MEMORY USED IN EMBEDDED SYSTEM CAN BE EITHER PROGRAM STORAGE MEMORY (ROM) OR DATA MEMORY (RAM).

MEMORY – PROGRAM STORAGE MEMORY

- STORES THE PROGRAM INSTRUCTIONS .
- RETAINS ITS CONTENTS EVEN AFTER THE POWER TO IT IS TURNED OFF. IT IS GENERALLY KNOWN AS NON VOLATILE STORAGE MEMORY.
- DEPENDING ON THE FABRICATION, ERASING AND PROGRAMMING TECHNIQUES THEY ARE CLASSIFIED INTO



MEMORY – PROGRAM STORAGE MEMORY

1. MASKED ROM (MROM)

- ONE-TIME PROGRAMMABLE MEMORY. USES HARDWIRED TECHNOLOGY FOR STORING DATA. THE DEVICE IS FACTORY PROGRAMMED BY MASKING AND METALLIZATION PROCESS ACCORDING TO THE DATA PROVIDED BY THE END USER
- THE PRIMARY ADVANTAGE OF MROM IS LOW COST FOR HIGH VOLUME PRODUCTION. THEY ARE THE LEAST EXPENSIVE TYPE OF SOLID STATE MEMORY
- THE LIMITATION WITH MROM BASED FIRMWARE STORAGE IS THE INABILITY TO MODIFY THE DEVICE FIRMWARE AGAINST FIRMWARE UPGRADES. SINCE THE MROM IS PERMANENT IN BIT STORAGE, IT IS NOT POSSIBLE TO ALTER THE BIT INFORMATION

MEMORY – PROGRAM STORAGE MEMORY

2. PROGRAMMABLE READ ONLY MEMORY (PROM) / (OTP)

- UNLIKE MROM IT IS NOT PRE-PROGRAMMED BY THE MANUFACTURER.
- PROM/OTP HAS NICHROME OR POLYSILICON WIRES ARRANGED IN A MATRIX, THESE WIRES CAN BE FUNCTIONALLY VIEWED AS FUSES.
- IT IS PROGRAMMED BY A PROM PROGRAMMER WHICH SELECTIVELY BURNS THE FUSES ACCORDING TO THE BIT PATTERN TO BE STORED.
- FUSES WHICH ARE NOT BLOWN/BURNED REPRESENTS A LOGIC “1” WHERE AS FUSES WHICH ARE BLOWN/BURNED REPRESENTS A LOGIC “0”.THE DEFAULT STATE IS LOGIC “1”
- OTP IS WIDELY USED FOR COMMERCIAL PRODUCTION OF EMBEDDED SYSTEMS WHOSE PROTO-TYPED VERSIONS ARE PROVEN AND THE CODE IS FINALIZED.
- IT IS A LOW COST SOLUTION FOR COMMERCIAL PRODUCTION. OTPS CANNOT BE REPROGRAMMED.

MEMORY – PROGRAM STORAGE MEMORY

3. ERASABLE PROGRAMMABLE READ ONLY MEMORY (EPROM)

- ERASABLE PROGRAMMABLE READ ONLY (EPROM) MEMORY GIVES THE FLEXIBILITY TO RE-PROGRAM THE SAME CHIP.
- EPROM STORES THE BIT INFORMATION BY CHARGING THE FLOATING GATE OF AN FET.
- BIT INFORMATION IS STORED BY USING AN EPROM PROGRAMMER, WHICH APPLIES HIGH VOLTAGE TO CHARGE THE FLOATING GATE.
- EPROM CONTAINS A QUARTZ CRYSTAL WINDOW FOR ERASING THE STORED INFORMATION. IF THE WINDOW IS EXPOSED TO ULTRA VIOLET RAYS FOR A FIXED DURATION, THE ENTIRE MEMORY WILL BE ERASED.
- EVEN THOUGH THE EPROM CHIP IS FLEXIBLE IN TERMS OF RE-PROGRAMMABILITY, IT NEEDS TO BE TAKEN OUT OF THE CIRCUIT BOARD AND NEEDS TO BE PUT IN A UV ERASER DEVICE FOR 20 TO 30 MINUTES

MEMORY – PROGRAM STORAGE MEMORY

4. ELECTRICALLY ERASABLE PROGRAMMABLE READ ONLY MEMORY (EEPROM)

- ERASABLE PROGRAMMABLE READ ONLY (EPROM) MEMORY GIVES THE FLEXIBILITY TO RE-PROGRAM THE SAME CHIP USING ELECTRICAL SIGNALS
- THE INFORMATION CONTAINED IN THE EEPROM MEMORY CAN BE ALTERED BY USING ELECTRICAL SIGNALS AT THE REGISTER/BYTE LEVEL
- THEY CAN BE ERASED AND REPROGRAMMED WITHIN THE CIRCUIT
- IT PROVIDES GREATER FLEXIBILITY FOR SYSTEM DESIGN
- THE ONLY LIMITATION IS THEIR CAPACITY IS LIMITED WHEN COMPARED WITH THE STANDARD ROM (A FEW KILOBYTES).

MEMORY – PROGRAM STORAGE MEMORY

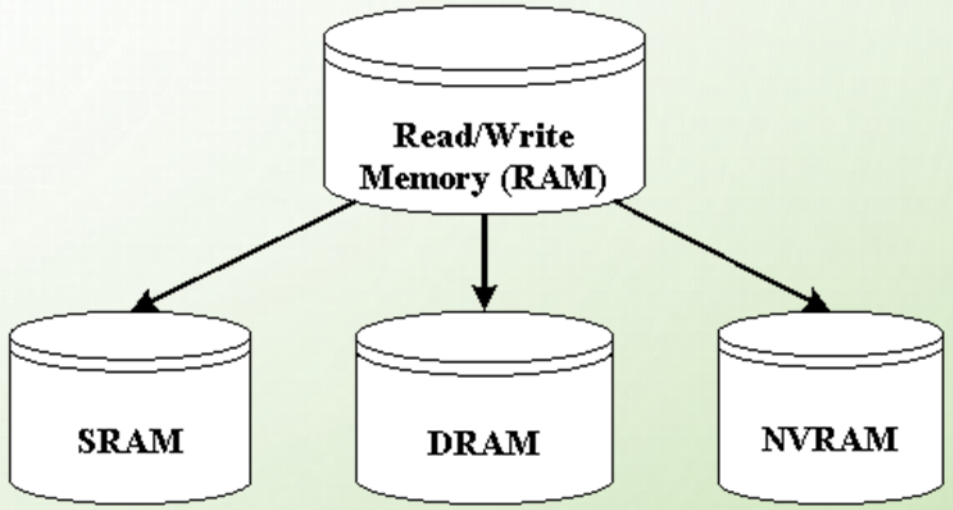
5. FLASH

- THIS MEMORY IS A VARIATION OF EEPROM TECHNOLOGY.
- IT COMBINES THE RE-PROGRAMMABILITY OF EEPROM AND THE HIGH CAPACITY OF STANDARD ROMS.
- FLASH MEMORY IS ORGANIZED AS SECTORS (BLOCKS) OR PAGES.
- FLASH MEMORY STORES INFORMATION IN AN ARRAY OF FLOATING GATE MOSFET TRANSISTORS.
- THE ERASING OF MEMORY CAN BE DONE AT SECTOR LEVEL OR PAGE LEVEL WITHOUT AFFECTING THE OTHER SECTORS OR PAGES.
- EACH SECTOR/PAGE SHOULD BE ERASED BEFORE RE-PROGRAMMING.

MEMORY – DATA MEMORY

READ-WRITE MEMORY/RANDOM ACCESS MEMORY (RAM)

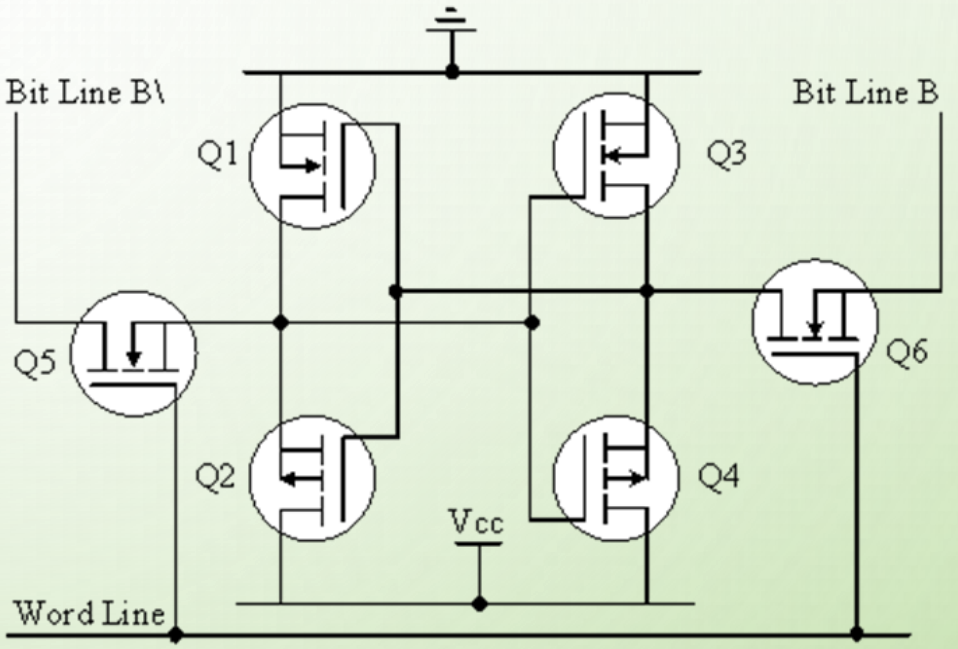
- RAM IS THE DATA MEMORY OR WORKING MEMORY OF THE CONTROLLER/PROCESSOR
- RAM IS VOLATILE, MEANING WHEN THE POWER IS TURNED OFF, ALL THE CONTENTS ARE DESTROYED
- RAM IS A DIRECT ACCESS MEMORY, MEANING WE CAN ACCESS THE DESIRED MEMORY LOCATION DIRECTLY WITHOUT THE NEED FOR TRAVERSING THROUGH THE ENTIRE MEMORY LOCATIONS TO REACH THE DESIRED MEMORY POSITION (I.E. RANDOM ACCESS OF MEMORY LOCATION)



MEMORY – DATA MEMORY

1. STATIC RAM (SRAM)

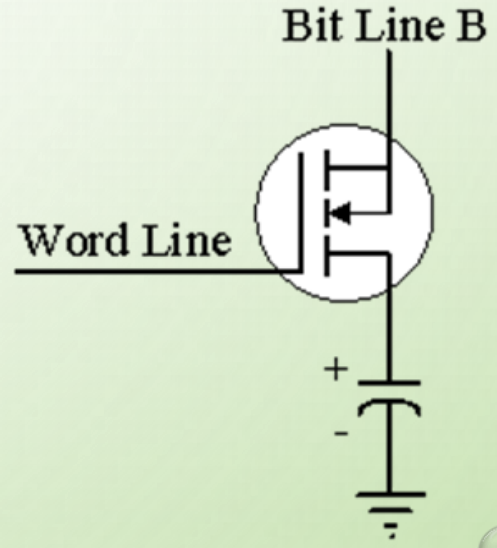
- STATIC RAM STORES DATA IN THE FORM OF VOLTAGE. THEY ARE MADE UP OF FLIP-FLOPS
- IN TYPICAL IMPLEMENTATION, AN SRAM CELL (BIT) IS REALIZED USING 6 TRANSISTORS (OR 6 MOSFETS). FOUR OF THE TRANSISTORS ARE USED FOR BUILDING THE LATCH (FLIP-FLOP) PART OF THE MEMORY CELL AND 2 FOR CONTROLLING THE ACCESS.
- STATIC RAM IS THE FASTEST FORM OF RAM AVAILABLE. SRAM IS FAST IN OPERATION DUE TO ITS RESISTIVE NETWORKING AND SWITCHING CAPABILITIES



MEMORY – DATA MEMORY

2. DYNAMIC RAM (DRAM)

- DYNAMIC RAM STORES DATA IN THE FORM OF CHARGE. THEY ARE MADE UP OF MOS TRANSISTOR GATES.
- THE ADVANTAGES OF DRAM ARE ITS HIGH DENSITY AND LOW COST COMPARED TO SRAM.
- THE DISADVANTAGE IS THAT SINCE THE INFORMATION IS STORED AS CHARGE IT GETS LEAKED OFF WITH TIME AND TO PREVENT THIS THEY NEED TO BE REFRESHED PERIODICALLY.
- SPECIAL CIRCUITS CALLED DRAM CONTROLLERS ARE USED FOR THE REFRESHING OPERATION. THE REFRESH OPERATION IS DONE PERIODICALLY IN MILLISECONDS INTERVAL.



SRAM and DRAM cell comparison

SRAM Cell	DRAM Cell
Made up of 6 CMOS transistors (MOSFET)	Made up of a MOSFET and a capacitor
Doesn't Require refreshing	Requires refreshing
Low capacity (Less dense)	High Capacity (Highly dense)
More expensive	Less Expensive
Fast in operation. Typical access time is 10ns	Slow in operation due to refresh requirements. Typical access time is 60ns. Write operation is faster than read operation.

MEMORY – DATA MEMORY

3. NON VOLATILE RAM (NVRAM)

- RANDOM ACCESS MEMORY WITH BATTERY BACKUP
- IT CONTAINS STATIC RAM BASED MEMORY AND A MINUTE BATTERY FOR PROVIDING SUPPLY TO THE MEMORY IN THE ABSENCE OF EXTERNAL POWER SUPPLY
- THE MEMORY AND BATTERY ARE PACKED TOGETHER IN A SINGLE PACKAGE
- NVRAM IS USED FOR THE NON VOLATILE STORAGE OF RESULTS OF OPERATIONS OR FOR SETTING UP OF FLAGS ETC.
- DS1744 FROM MAXIM/DALLAS IS AN EXAMPLE FOR 32KB NVRAM

CRITERIA FOR CHOOSING A MICROCONTROLLER

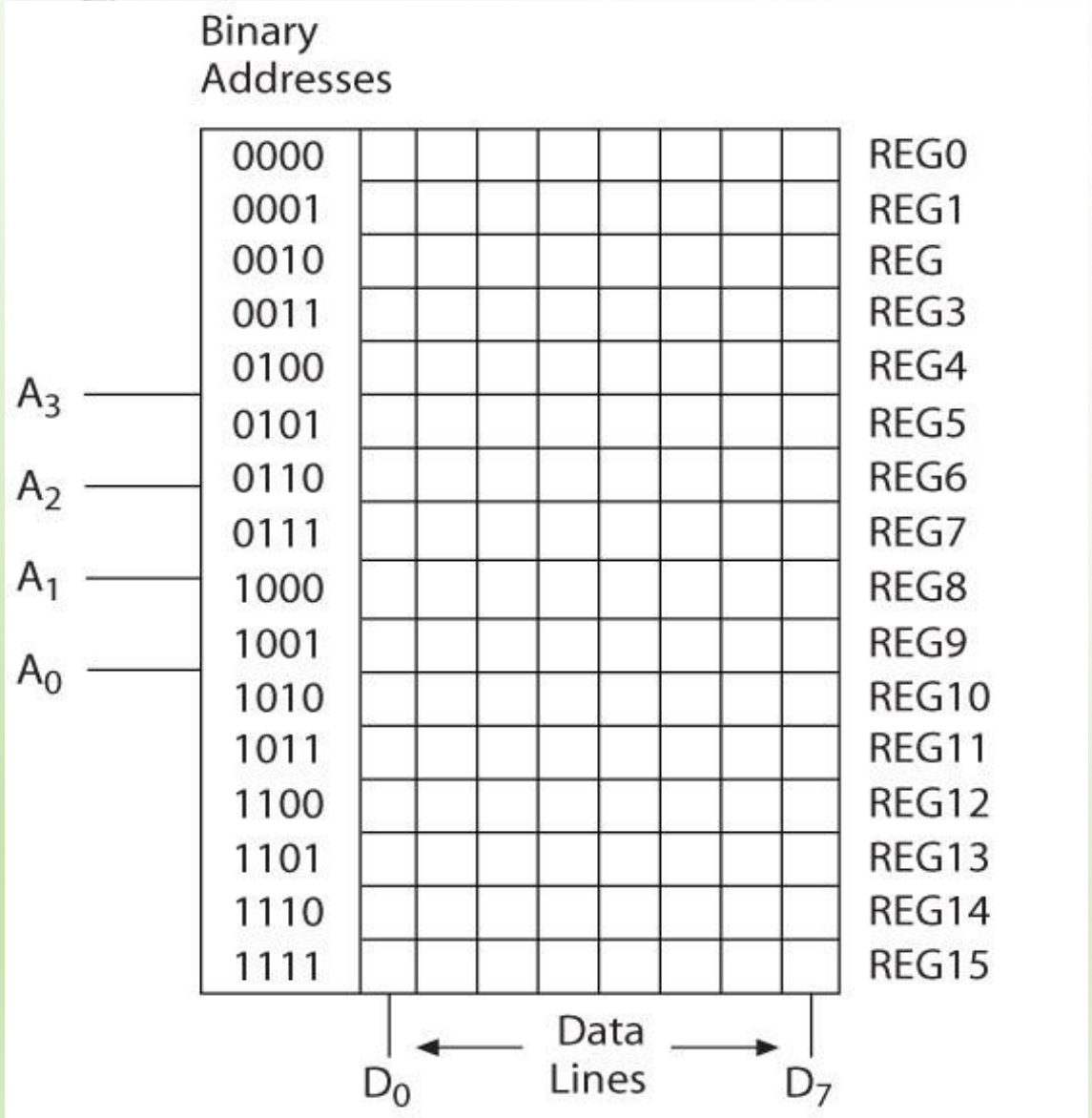
- 1. MEETING THE REQUIREMENTS –
  - - SPEED
  - - PACKAGING
  - - POWER CONSUMPTION
  - - RAM & ROM CAPACITY ON CHIP
  - - EASE OF UPGRADATION
  - - COST PER UNIT
- 2. AVAILABILITY FOR DEVELOPMENT TOOLS(ASSEMBLERS, DEBUGGERS, IDE) FOR THE MC
- 3. AVAILABILITY IN MARKET AND DURATION OF AVAILABILITY.

WHAT DOES MEMORY LOOK LIKE ?

- RAM, ASSOCIATED WITH TWO NUMBERS
  - ADDRESS AND DATA
  - ADDRESS IS THE LOCATION
  - DATA IS THE ACTUAL VALUE
- MEMORY STORES BOTH **DATA** AND **MACHINE INSTRUCTIONS**
- **EVERYTHING STORED IN MEMORY IS IN BITS**

Address	Data
0	36
1	3765
2	786
3	356
4	252
5	67980
6	2355
7	4234
8	3466

RAM



- Data and code (instructions) are represented using bits
- Memory stores both data and code
- 16 bytes of memory shown in the diagram
- What is the capacity if I double the address bits?
- What is the capacity if I double the data bits?

REPRESENTING NUMBERS

- UNSIGNED INTEGERS
  - ALL EIGHT BITS REPRESENT THE MAGNITUDE OF A NUMBER
    - BIT-*7* TO BIT-*0*
  - RANGE 00_H TO FF_H (0_{10} TO 255_{10})
- SIGNED INTEGERS
  - 2'S COMPLEMENT
    - MSB (HERE BIT-*7*) IS SIGN BIT
  - POSITIVE NUMBERS: 00_H TO $7F_H$ (0_{10} TO 127_{10})
  - NEGATIVE NUMBERS: 80_H TO FF_H (-1_{10} TO -128_{10})

REPRESENTING NUMBERS

- BINARY CODED DECIMAL NUMBERS (BCD)
  - 8-BIT NUMBER DIVIDED INTO TWO GROUPS OF FOUR
    - EACH GROUP REPRESENTS A DECIMAL DIGIT FROM 0 TO 9
  - A_H THROUGH F_H ARE INVALID
  - EXAMPLE: $0010\ 0101_{BCD} = 25_{10}$