INTRODUCTION TO EMBEDDED SYSTEMS

MR. UMESH KOYANDE

MR. KIRAN DATAR

Core Of The Embedded System

General purpose and domain specific processors

- Microprocessors
- Microcontrollers
- Digital Signal Processors

Application Specific Integrated Circuits (ASICs)

Programmable Logic Devices (PLDs)

Commercial off-the-shelf Components (COTS)

MICROPROCESSOR	VS.	MICROCONROLLERS
A silicon chip representing a central processing unit (CPU)		A chip that contains a CPU, RAM, register arrays, on chip ROM / FLASH memory, timer and interrupt control units and I/O ports.
Doesn't contain a built in I/O port		Multiple built-in I/O ports
It is a dependent unit		It is a self–contained unit
Limited power saving options		Includes lot of power saving features
It has few bit manipulation instructions		It has many bit manipulation instructions

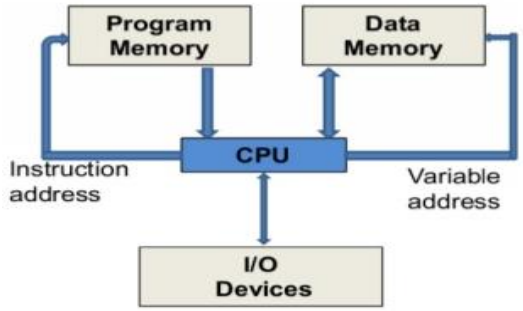
DIGITAL SIGNAL PROCESSORS

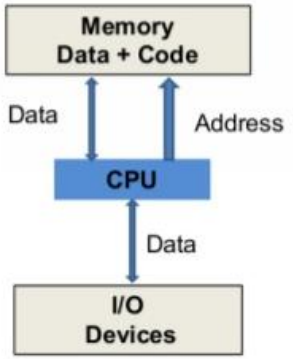
- 8/16/32 bit microprocessors designed specifically to meet the computational demands and power constraints of today's embedded audio, video, and communications applications
- A typical digital signal processor incorporates the following key units :
  - **Program memory**
  - **Data Memory**.
  - **Computational Engine**.
  - **I/O Unit**

RISC	VS.	CISC
Lesser number of instructions		Greater number of Instructions
Instruction pipelining		Generally no instruction pipelining feature
Orthogonal instruction set		Non-orthogonal instruction set
A large number of registers are available		Limited number of general purpose registers
Single, fixed length instructions		Variable length instructions
With Harvard Architecture		Can be Harvard or Von-Neumann Architecture

HARVARD ARCHITECTURE VS. VON NEUMANN ARCHITECTURE

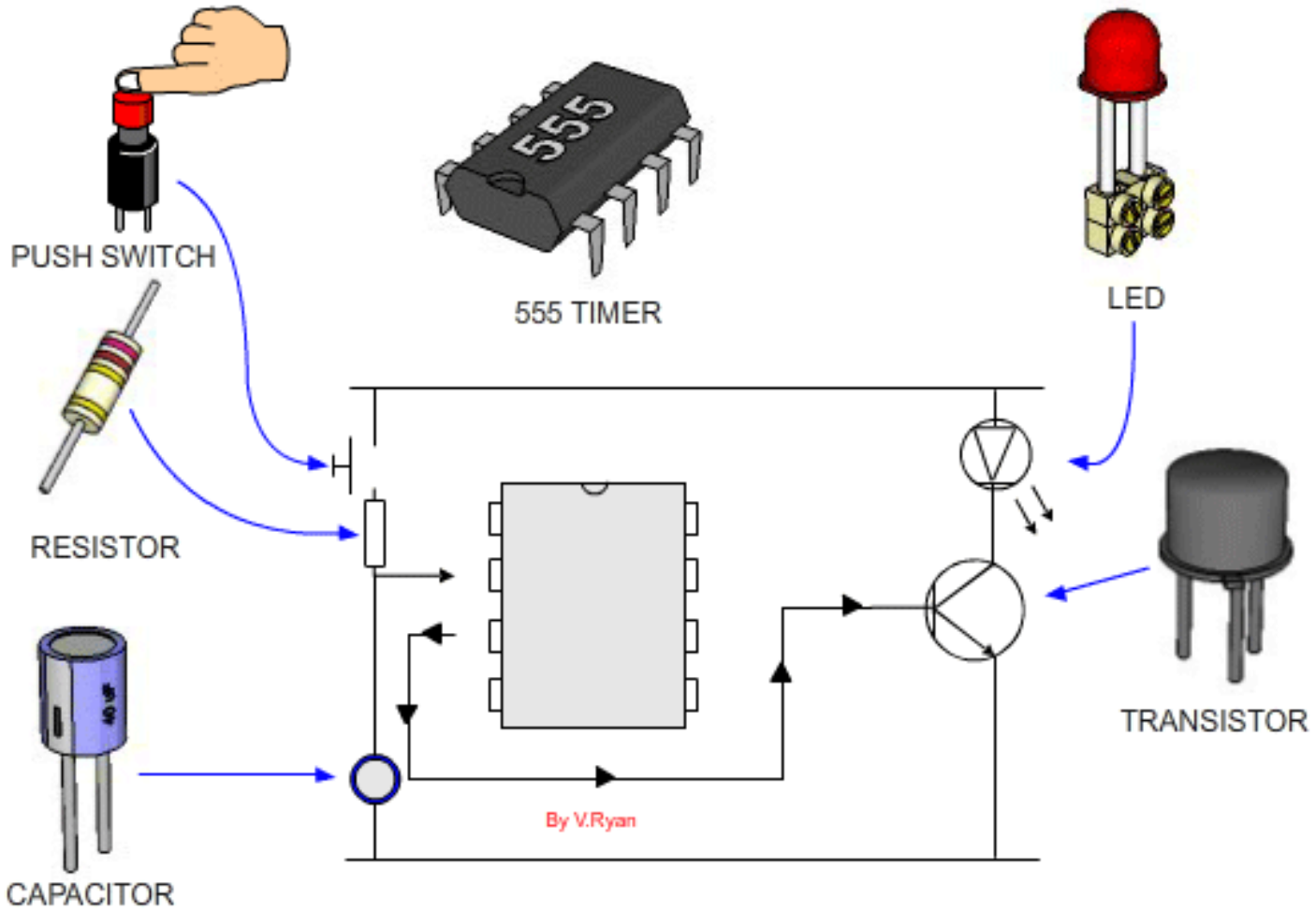
HARVARD ARCHITECTURE	VON NEUMANN ARCHITECTURE
Separate buses for instruction and data fetching	Single shared bus for instruction and data fetching
Easier to pipeline, so high performance can be achieved	Low performance compared to Harvard architecture
Comparatively high cost	Comparartively Cheaper
No chances for accidental corruption of program memory	Chances for accidental corruption of program memory

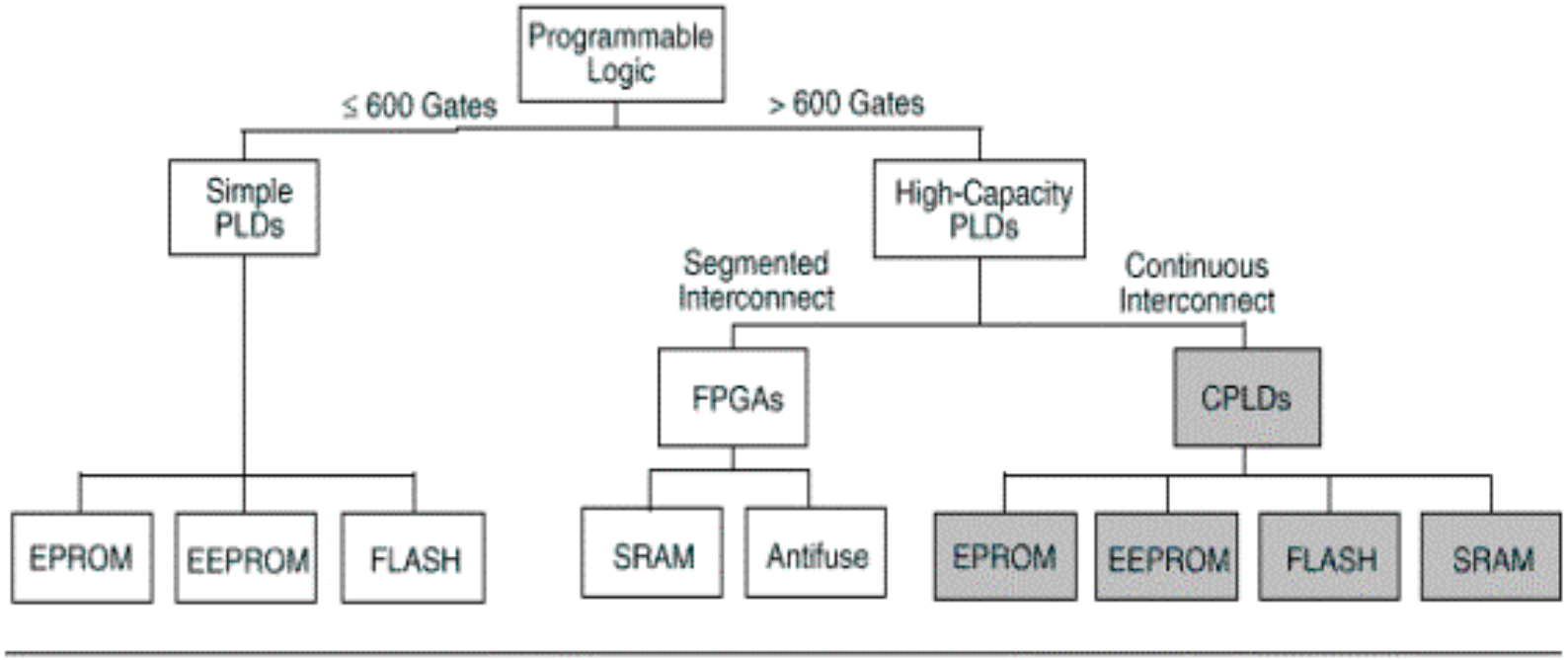




APPLICATION SPECIFIC INTEGRATED CIRCUITS (ASIC)

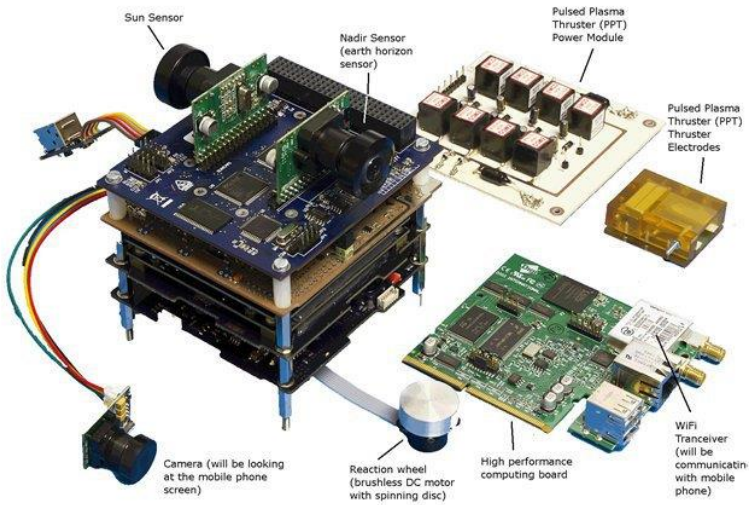
- Integrates several functions into a single chip
- Reduces the system development cost
- Profitable only for large volume commercial productions
- Non-Recurring Engineering – Fabrication requires a non-refundable initial investment





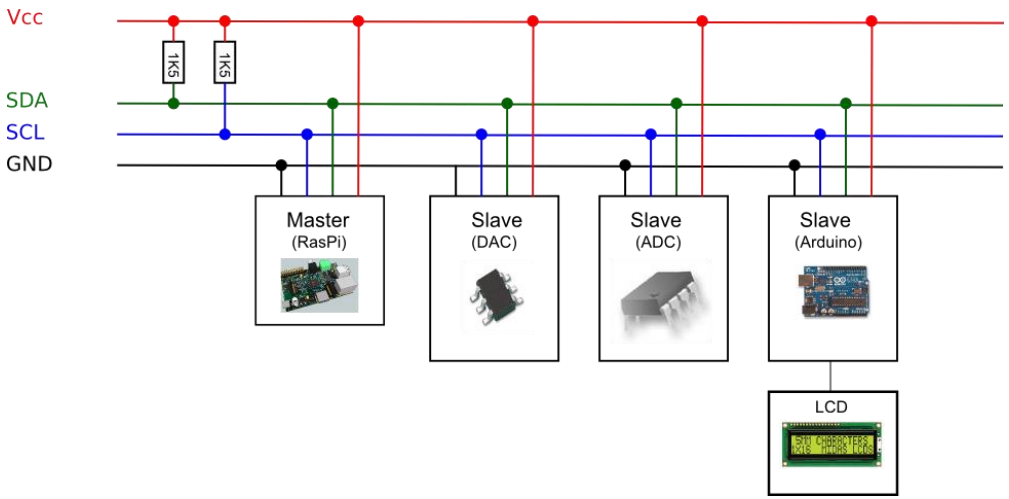
PROGRAMMABLE LOGIC DEVICES

COMMERCIAL OFF-THE-SHELF COMPONENTS (COTS)



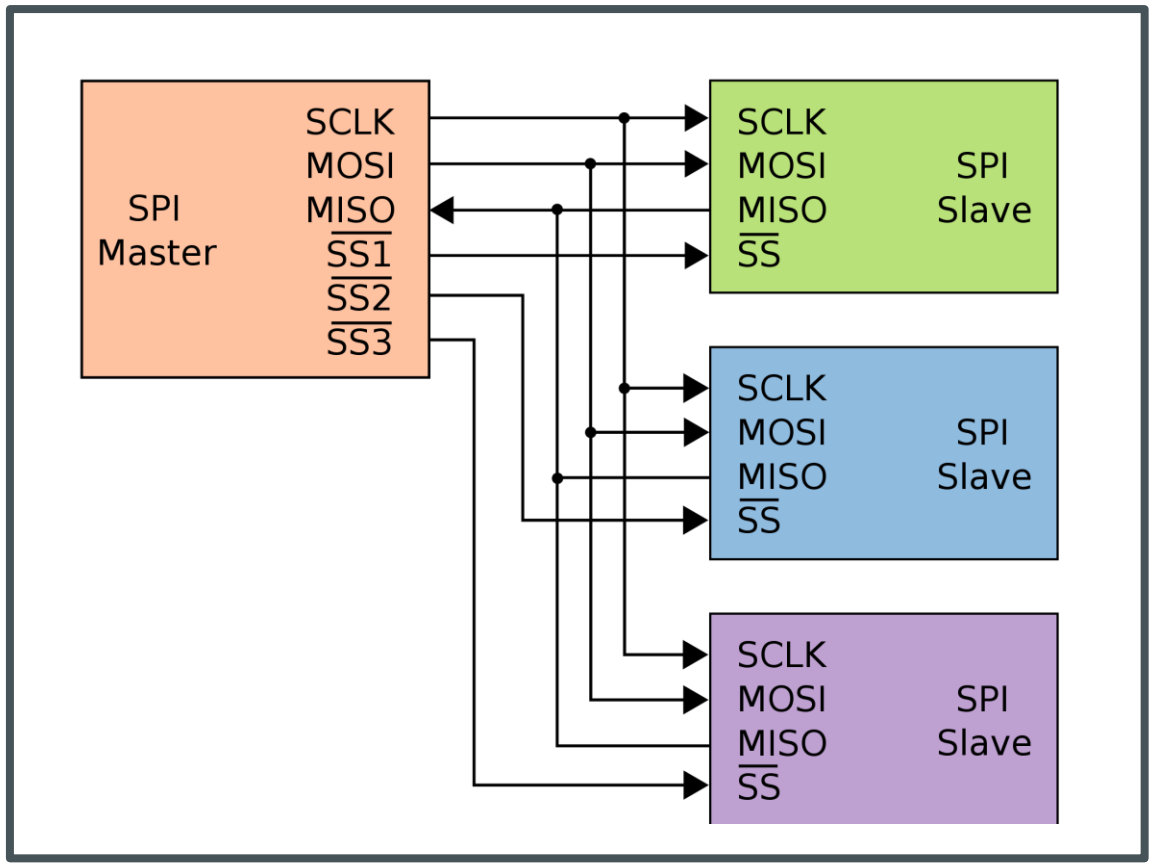
- Designed to provide easy integration and interoperability with existing system components
- Readily available in the market, are cheap and development time to a great extent
- Reduces the time to market
- No operational and manufacturing standards
- Hardware plug-in and firmware interface compatibility varies with vendors

ONBOARD COMMUNICATION INTERFACES



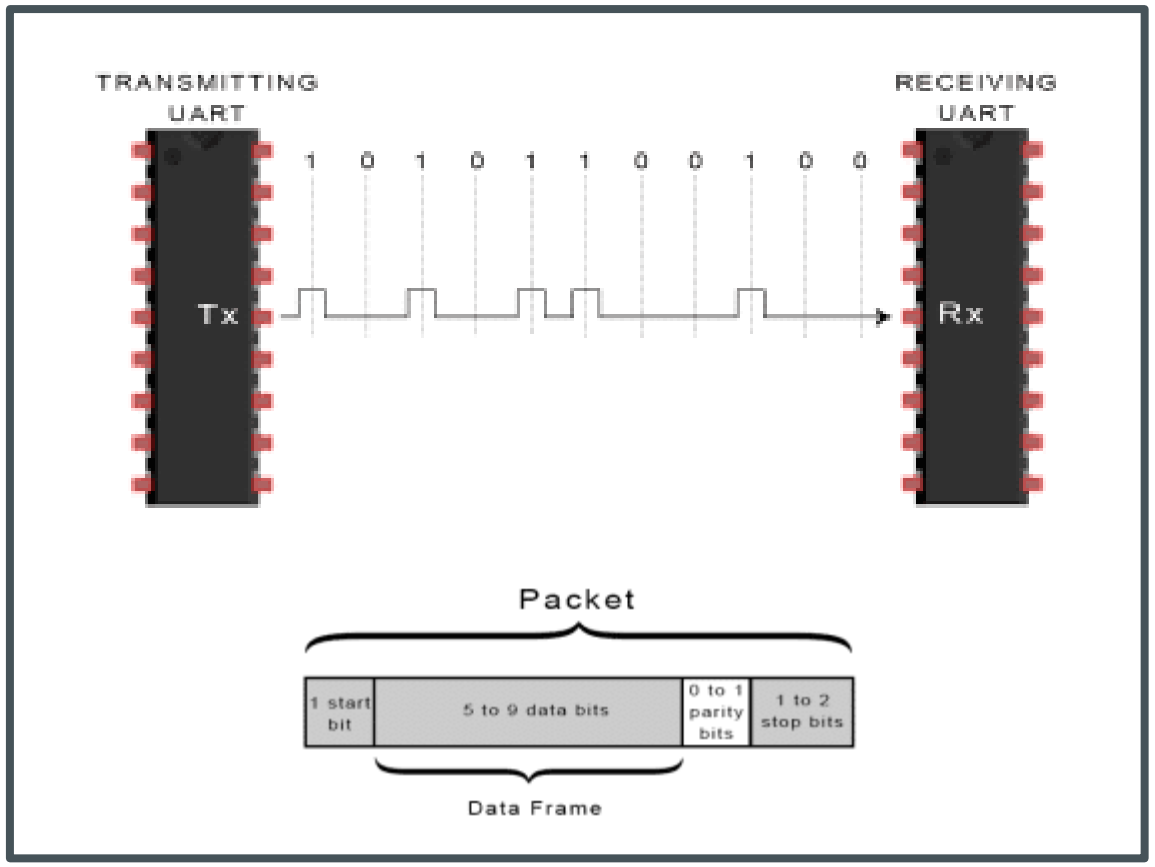
- Inter Integrated Circuit Bus (I2C) Bus
  - I2C bus comprise of two bus lines
    - **Serial Clock–SCL**
    - **Serial Data–SDA**
  - Devices connected to the I2C bus can act as either ‘Master’ device or ‘Slave’ device
  - ‘Master’ – initiating/terminating data transfer, sending data and generating necessary synchronization clock pulses.
  - ‘Slave’ devices wait for the commands from the master and respond upon receiving the commands
  - Bi-directional data transfer

ONBOARD COMMUNICATION INTERFACES



- Serial Peripheral bus
  - Synchronous bi-directional full duplex bus
  - Single master multi-slave system
  - SPI requires four signal lines for communication
    - **Master Out Slave In (MOSI)**
    - **Master In Slave Out (MISO)**
    - **Serial clock (SCLK)**
    - **Slave Select (SS)**
  - Most suitable for applications requiring transfer of data in ‘streams’
  - SPI doesn’t support an acknowledgement mechanism

ONBOARD COMMUNICATION INTERFACES



- Universal Asynchronous Receiver Transmitter (UART)
  - Asynchronous form of serial data transmission
  - Communication settings for both transmitter and receiver should be set as identical
  - 'Start' bit informs the receiver that a data byte is about to arrive
  - If parity is enabled for communication, the UART of the transmitting device adds a parity bit
  - Receiving device discards the 'Start', 'Stop' and 'Parity' bit from the received bit stream and converts the received serial bit data to a word