UNIT: IV:

SERIAL COMMUNICATION INTERFACE

INTEL 8251 USART

USART - Universal Synchronous Asynchronous Receiver Transmitter

- Programmable chip designed for synchronous and Asynchronous serial data transmission
- Hardware peripheral used in microcontroller and microprocessor to enable serial communication

Parallel - Bits are sent simultaneously over its own separate wire

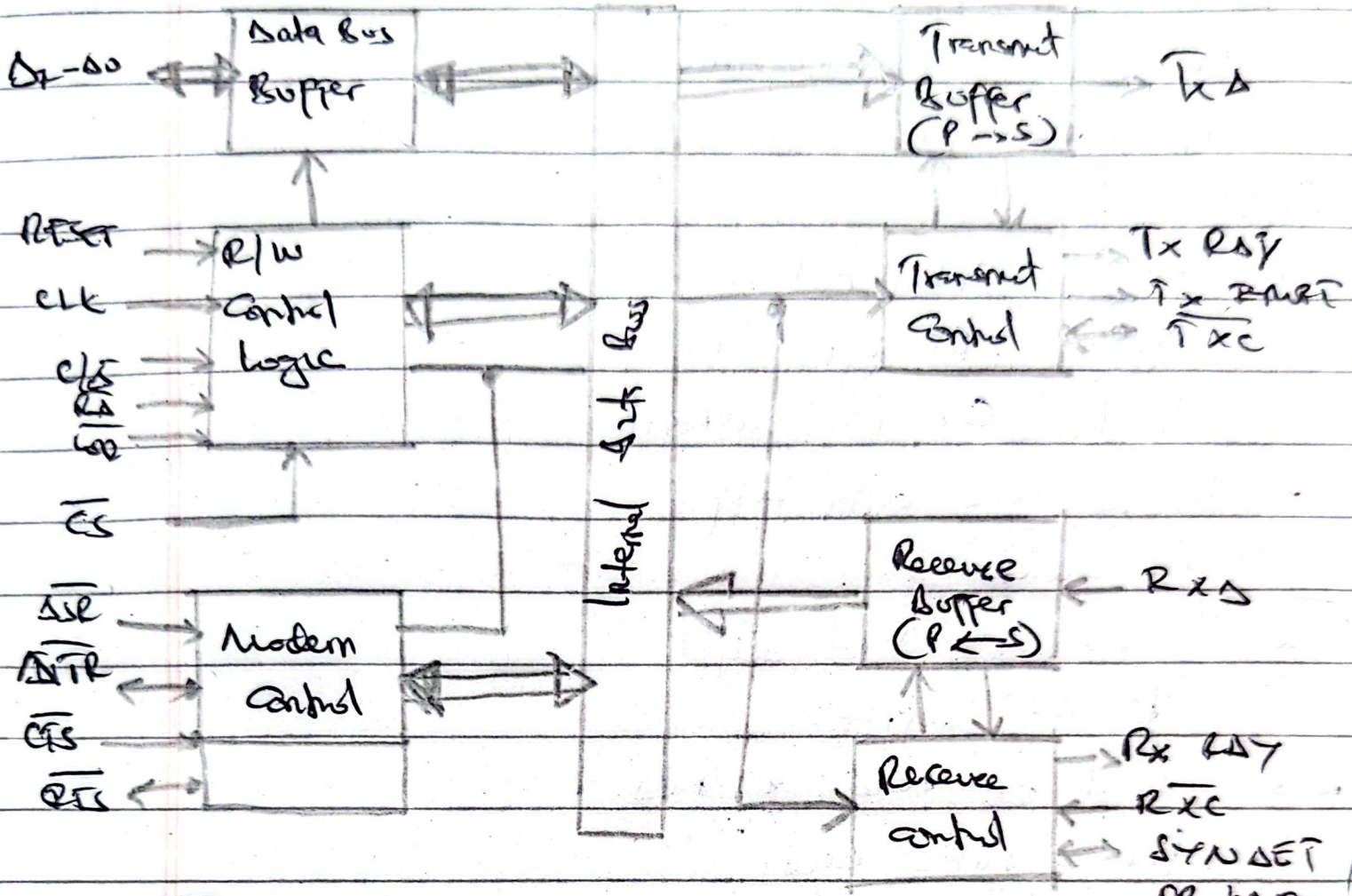
Serial Communication

- Means data is transmitted one bit at a time over a single wire (or single data line)
- Has 28 pin DIP.
- It receives a serial stream of bits and convert it into parallel data bytes to be read by a microprocessor.

USART works internally

1. Externally (with device)...
   - Communication is done serially (one bit at a time)
2. Internally. Inside the microcontroller)
   - Data is handled in parallel

USART BLOCK DIAGRAM.



Five Section in USART Block Diag

1. Input Signals
2. Transmitter Section
3. Receiver Section
4. Signals Associated with Receiver Section
5. Signal Associated with Modem Control

Input Signals

$\overline{CS}$ - Chip Select
- when is low 8251 selected by MPU for comm.

C/$\overline{D}$ - Control / Data
- used to control data when the signal is high CE is addressed
- when is low data buffer is addressed

$\overline{WR}$ - used to send output to the buffer

$\overline{RD}$ - used to read a status from status register

RESET - Reset

CLK - clock

data out to another device over serial comm.

USART Transmission Section
- Accept parallel data from microprocessor into Serial

Components Explanation.

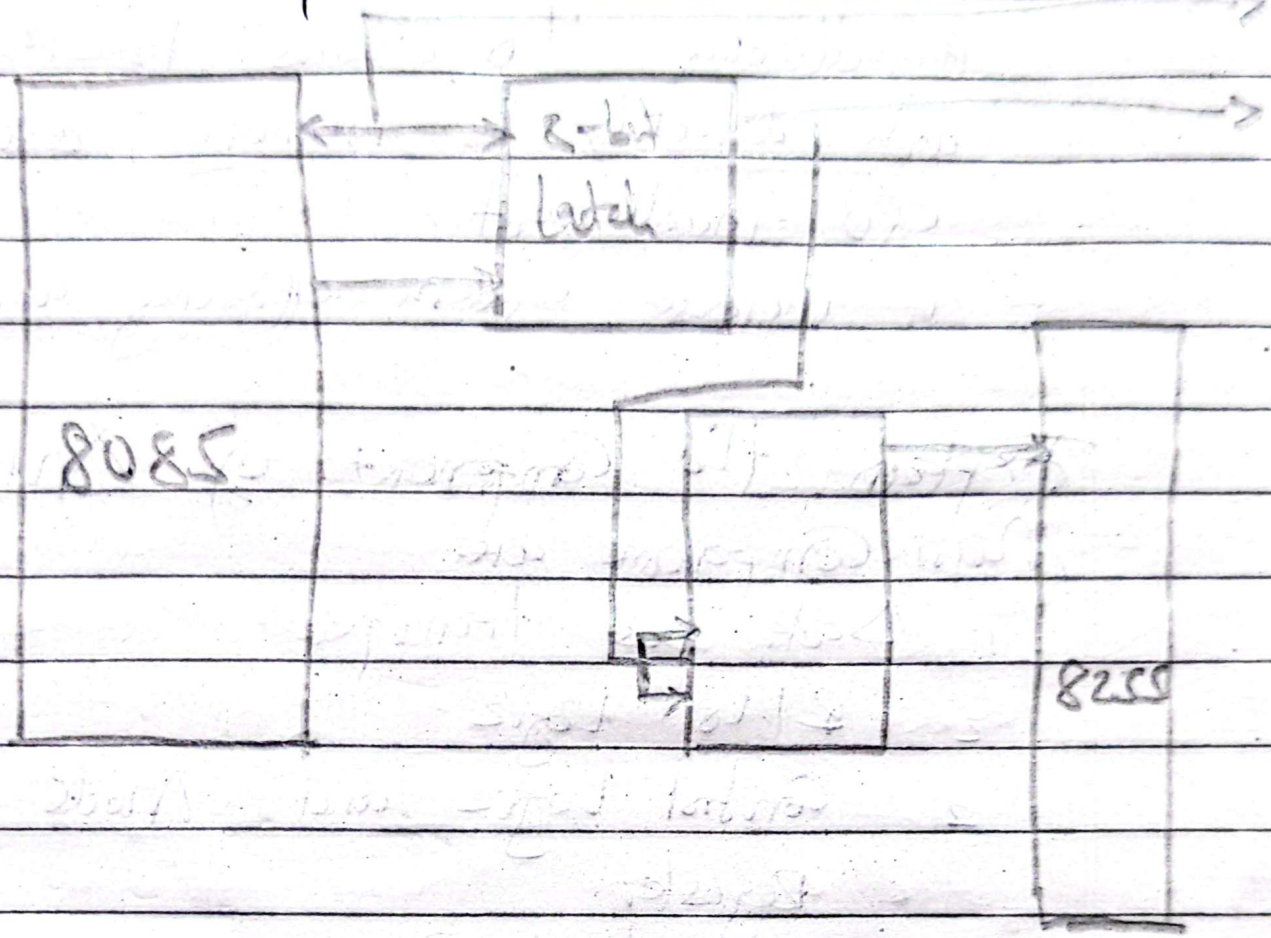
1. Data bus buffer
   - Helps in interfacing internal data bus to the system data bus
   - Temporarily stores data & controls the flow of data thru different part of system.
2. Modem control
   - Converts analog signals to digital signal
   - Helps computer to communicate over telephone line
   - Have the following pin.
     - DSR
     - DTR
     - CTS
3. Read/Write control Logic
   - used to control the overall working by selecting the operation (Read or write).
     - RD, WR, RESET, CLK. $\overline{CS}$
4. Transmitter buffer
   - This block is used for parallel to serial converter, that receive parallel data from CPU and convert it into serial signal (TXD)

S Transmit Control
- used to control the data transmission with the help of the following pin
  - TXRDY
  - TXEMPTY
  - TXE

B Receive buffer
- Act as buffer for the received data RXD.

I Receive Control
- used to control the receiving data
  - RXRDY
  - RXT
  - SYNDET/BD

Interfacing of 8255 (PPI) with 8085 processor.



DMA Controller

DMA - Direct Memory Access

- It allowing I/O devices to transfer data directly to memory, and relase CPU involvement
- It increase system efficiency and speed

Explain the Component of DMA

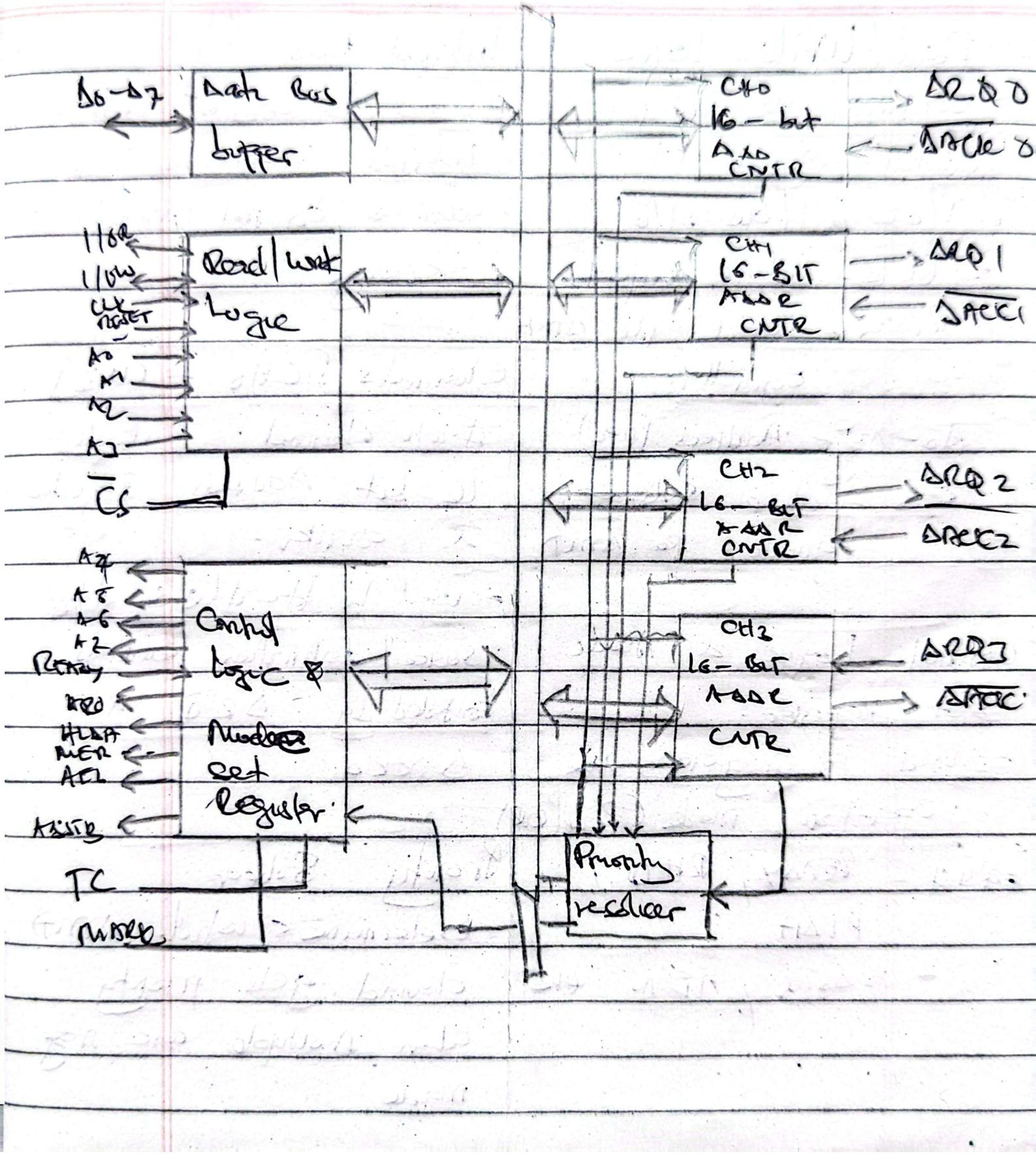
Main Components are

1. Data bus transfer
2. R/W Logic
3. Control Logic and Mode Set Register.
4. Internal bus
5. Channels
6. Priority Resolver.

① Data bus transfer.

- Connect DMA to the system data bus (D0-D7).
- Temporary storage for data during transfers btw memory & I/O devices

Internal bus



0756573018

2. Read/Write Logic
- Contain the interpret control signal like
  - I/OR, I/OW : I/O
  - Clk - clock input
  - RESET - Reset the DMA controller.
- A0-A3 - Address input
  CS - chip select
  Activate the DMA

Control Logic & Mode Set Register:
- used to configures the operation mode (Read/write)
- consist - DEMAY, RHO, HLDA
  - ASSTB, AEN etc

Internal Bus
- used to connect all internal components for data & control signal exchange

Channels [CH0 to CH3]
- Each channel include 16-bit Address Register & counter:
- used to handles the source/destination address controlled by DRQs & DACKs.

Priority Solver
- Determines which DMA channel gets priority when multiple are ready mode

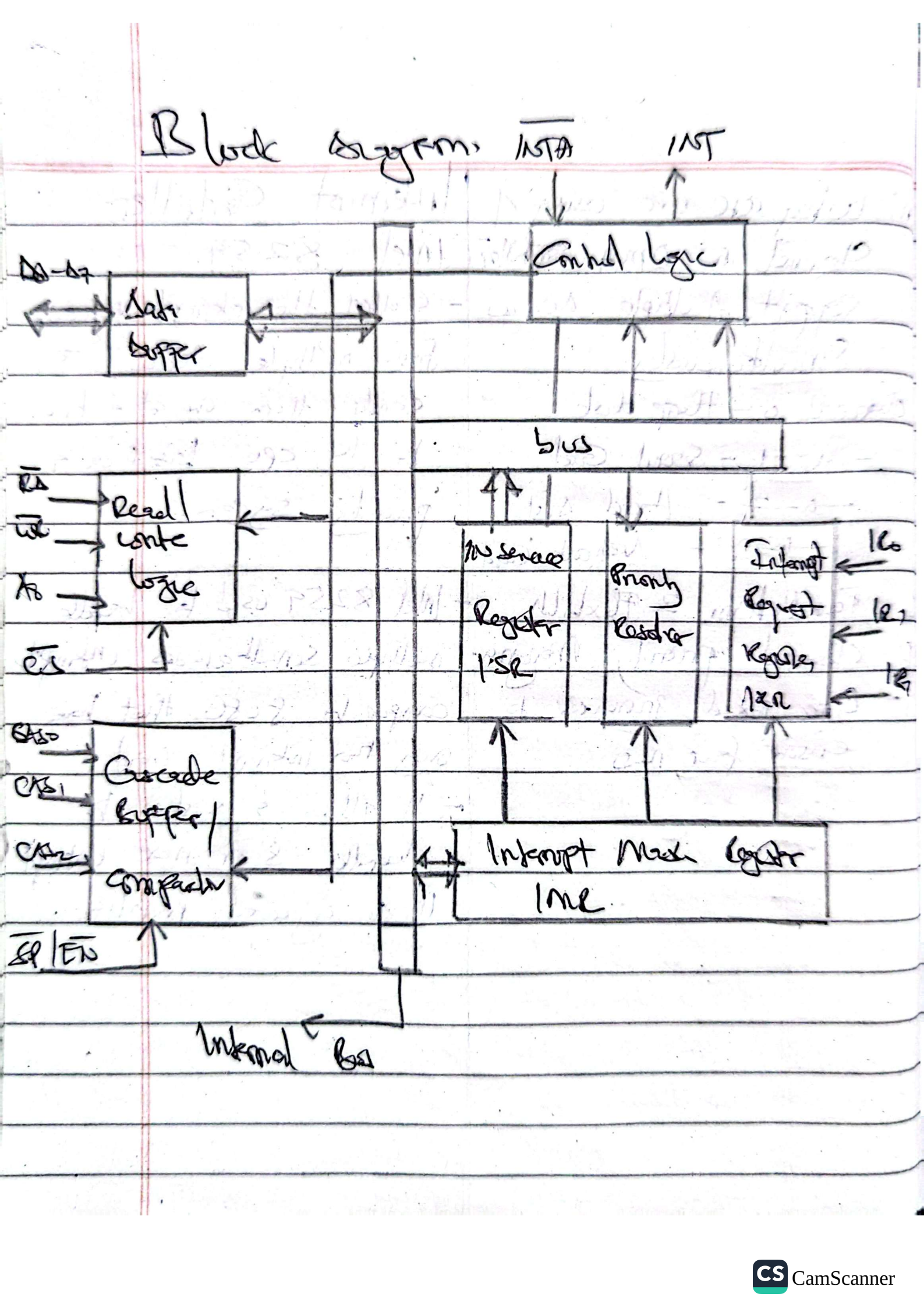
Qn: Why are we using 4 channel in DMA controller

1. Support multiple devices simultaneously
channel 0 - Floppy disk
channel 1 - Sound Card
channel 2 - Hard disk
channel 3 - Network Card
2. Parallelism & flexibility
3. channel priority management
- Give fixed priorities to ensure fair use

Interrupt Controller
Intel 8259.

- Collect the interrupt requests from multiple devices & sends them one at a time to the CPU based on a priority system
- Intel 8259 used to handle multiple simultaneous interrupt common in 8086 that have only two interrupt input
- It allows a system to handle 8 or more interrupts in an organized, prioritized

Block diagram:



Purpose

- Help microprocessor handle multiple hardware interrupt requests by prioritizing them

Block diagram Explanation -

1. IR0 to IR7 [Interrupt Request Lines]
   - Input lines where external devices send interrupt requests
   - And can handle 8 interrupt sources
2. IRR
   - used to store all incoming pending interrupt requests
3. ISR
   - It store all the interrupt requests those are being served.

Control Logic

- Handles communication between CPU and the PIC
- Sends INT to CPU and receives INTA from CPU

Priority Resolver

- Determines which interrupt request has a highest priority

R/W Logic

- used to Read the status, write commands
- Control signals (RD, WR, CS, A0)

Cascade Buffer

- used when connecting multiple 8259 in cascade.
- used to store and compare

Keyboard / Display Controller

INTEL 8279.

- keyboard and display controller designed to interface keyboard & 7-segment LED display with a microprocessor
- used to interfacing keyboard and display devices to 8085 or 8086 microprocessor.

Features

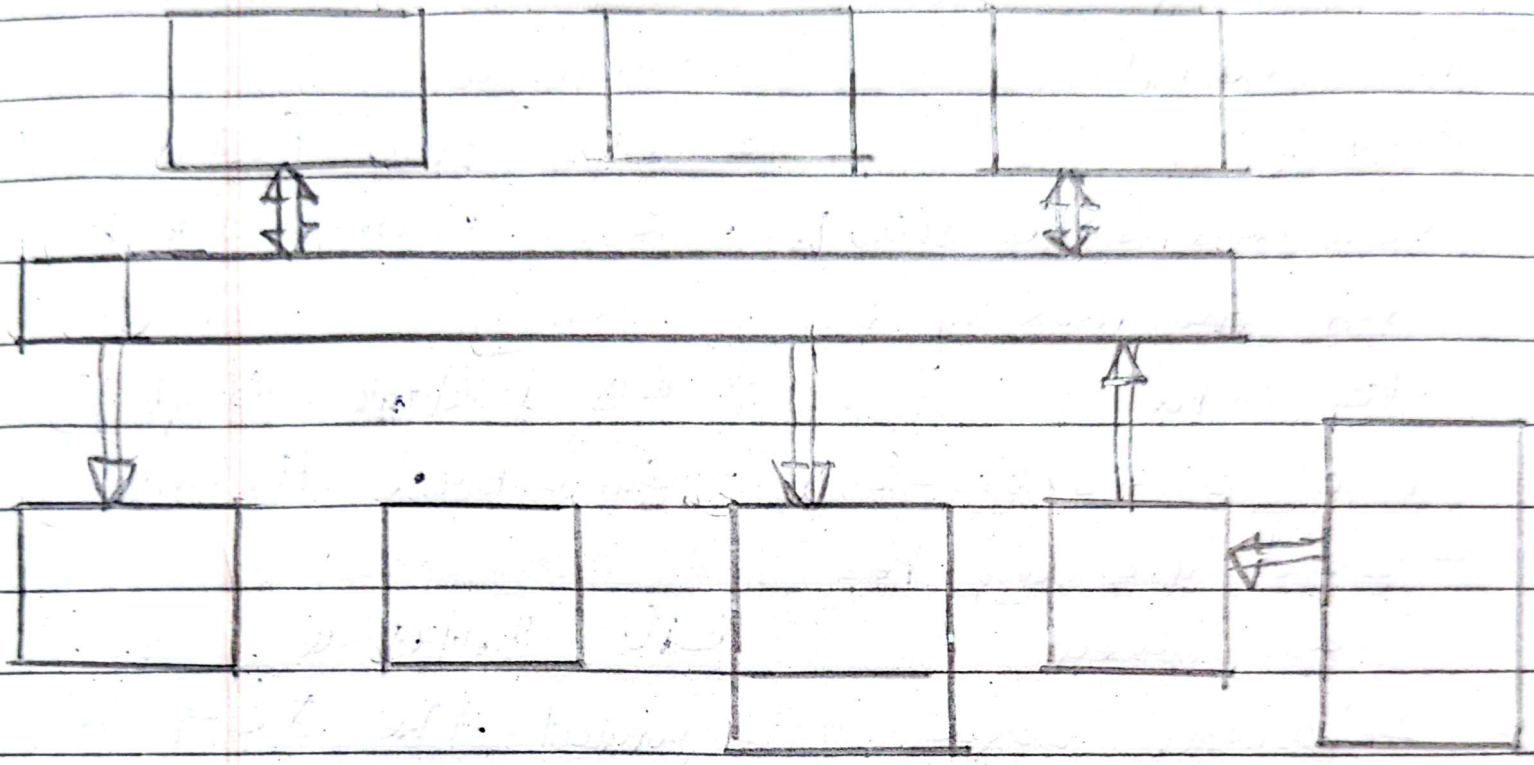
1. FIFO buffer.
   - 8-character keyboard
2. Keyboard Interface
   - scan a matrix keyboard,
   - handles key debouncing
2. Display Interface
   - can drive up to 16-character display
3. Mode
   - Scanned Sensor
   - Scanned keyboard

Four sections in keyboard display.

1. Keyboard Section
2. Display section.
3. Scan Section.
4. CPU Interface Section.

1. Keyboard Section
   - Handles keyboard scanning, debouncing and buffering key to codes
2. Display Section.
   - Manage display RAM and drives output device like LEDs
3. Scan section
   - Scan signal for both keyboard & display multiplexing

CPU Interface section - Communicates with the microprocessor for data transfer and control.



Components Explained

1. Keyboard Section
- Detects keypresses from matrix keypad.
Consist of:
- Scan Lines (SL0-SL2) to scan one row of a key matrix
- Return Lines (RL0-RL7)
- Used if a key has been pressed.
- Debouncing Logic
- Filters bouncing signals from mechanical switches
- FIFO buffer
- Store valid key code so CPU can read it later.

Display Section.
- Display RAM
- Hold the characters to be displayed.
- Output Lines (OUT0-OUT7)
- Send segment data to display.
- RD (Read display)
- Can blank the display

CPU Interface
- Connect the 8279 to the microprocessor. (CS to RESET)
- Data bus (D0-D7)

- - . . .

A/D Interfacing

- Analog to digital Interfacing
- Process of connecting a Analog-to Digital converter [ADC] to a microcontroller computer a digital byte

ADC 0804

- is 8-bit Analog-to-Digital Converter that converts analog voltage [0-5V] into a digital value

* Steps for converting

D/A Interfacing

- The digital-to-analog converter used to convert digital pulses to analog signals

Two ways of Creating a DAC

1. Binary weighted
2. R-2R Ladder.

① Weighted Resistor.

- Each digital bit controls a switch connected to a resistor, resistor

2. R-2R ladder

- Uses only two resistor value R & 2R arranged in ladder

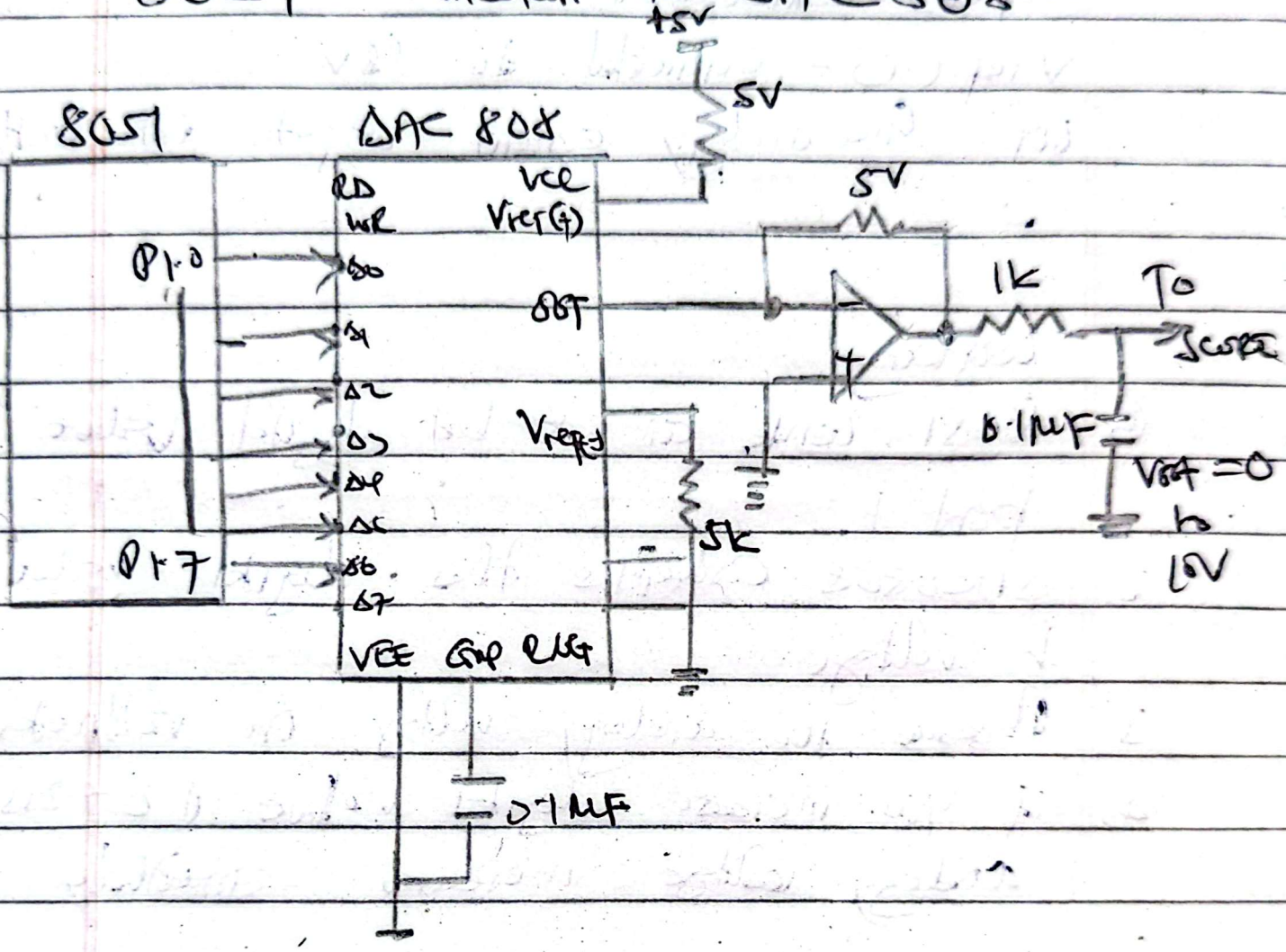
Explanation:

DA

1. Microcontroller (8051)

- Port 1 of Pro-Pin 7 - is used to send an 8-bit digital value to DAC
- 8 line are connected directly to D0-D7 pins of the DAC 0808.

8051 Connection to DAC808



DAC0808

D0-D7 – Receive the 8-bit digital input from the 8051

VCC – Connect to +5V power.

VEE – Negative voltage supply for internal circuit.

Vref(+) - connected to GND

V ref (+) - Connected to +5V

OUT - Gives analog current output or voltage

- - - -

Working:

1. ADC0808 takes an 8-bit digital value to port 1
2. ADC0808 converts the digital value to voltage
3. Observe the analog voltage on voltmeter
4. If you increase digital value (0-255) analog voltage increases smoothly

LCD INTERFACING

LCD - Liquid Crystal display
- Flat-panel display retro
Lgy used in electronic devices.

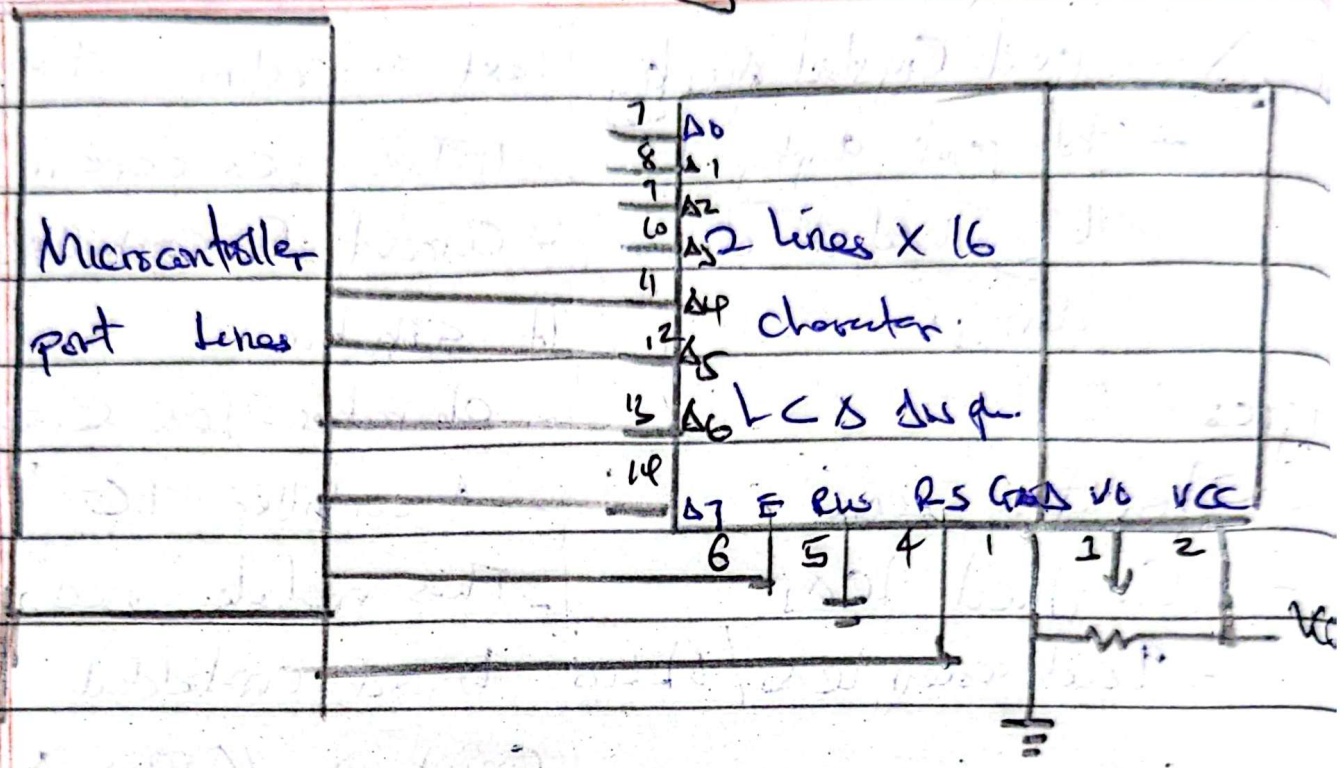
Types
- Character LCDs.
- Graphical LCDs
- Touch screen LCDs / OLEDs

① 16 x 2 character LCDs
- It's alphanumeric display module used in embedded system.
- It displays 16 characters per line on 2 line.

Character LCD Module (16x2) - Is a ready-to use hardware component that allows microcontroller to display text or custom characters
- Include LCD screen
- Circuit [controller] it contains.
- character LCD screen.
- A controller IC
- This module used in 8051 based embedded project
Consist of 16 rows x 2 columns
- it has 16 pin.

Component →
① LCD panel
- Where character are shown.
② Backlight (LEDs)
- Help readability
3. Pin header [16 pins] - for connecting to microcontroller
4. Contrast control Pin

Circuit diagram: Interfacing 16x2 LCD with



Hardware Configuration of LCD with 8051 / 8086 / 8085

① LCD initialization

- LCD can operate in 8-bit or 4-bit Methods either differs depends on one your using 8-bit.

1st - Function set 8 bits, 2 Line

- Display ON, Cursor ON
- Entry Mode set
- Clear display & return cursor

Sending Data to LCD

- In order to display, you need to send data to the LCD using microcontroller

Eg:
Command / RS=0 - Configure LCD (clear, cursor on/off)
Data RS=1 - Display character on screen.

Steps in Sending Data

1. Set RS = 1 (data mode)
2. Set RW = 0 (write mode)
3. Send the character code
4. Toggle Enable (E) from 0 → 1
5. Wait for the LCD to process
6. Repeat for sending another data

External Memory Interfacing (READ or WRITE)

- When built-in memory of microcontroller is not enough, process allows processor to read from or write to external memory chips

Access to External Memory in 8051

In 8051
- Port 0 act as data bus sending PCL.
- Port 2 - Send the PC address to external memory.
- ALE - Flag on external latch to store PCL
- Port 0 - Then switches function & become data bus receiving PC high from

SERIAL COMMUNICATION

Serial Communication

- Method of transmitting data one bit at a time over a single communication line or channel.
- used to send data b/w microcontroller, sensors, computer etc

Types
- Asynchronous
- Synchronous

Asynchronous
- does not share clock signal

Two common serial communication Standard
1. UART
2. SPI

1. UART Eg: Arduino to Computer
- Universal Asynchronous Receiver Transmitter
- Data is transmitted without a clock signal, using start and stop bits in data frame.

SPI
- Serial Peripheral Interface known as SS and
- This is a synchronous serial communication protocol, uses clock signal for synchronous.
- uses four lines & support full duplex

Adv. of serial over Parallel
- Serial requires fewer wires, reduce complexity and cost of connection

I^2C Bus
- Inter-Integrated Circuit
- is a two-wire synchronous serial communication used to connect multiple low-speed devices in microcontroller
- Short-distance comm.
Eg Arduino to Temperature sensor

LCD display
- Liquid Crystal display
- used to display character, numbers and symbols

- used as output device in microprocessor system to show information
- Low power consumption
- Can display character & graphics

SPI - Purpose
- Serial communication protocol used to transfer data b/w microprocessor [master] and peripheral devices [slaves]
- Full duplex data trans

4 main signal lines
- SCLK
- MOSI
- MISO
- SS

SCLK
- Serial clock.
- Clock signal generated by master to synchronize data transfer.
- MOSI [Master Out Slave In]
  - Line for data sent from master to slave
- MISO
  - Master In Slave Out]
  - Line for data sent from slave to master
- SS
  - Slave Select
  - used by master to select the slave devices.

Adv of SPI over I^2C
- Faster Data transfer
- No addressing, reduce complexity

EXTERNAL COMMUNICATION INTERFACE

RS232.

RS 232

- Stands for Recommended ~~232~~ Standard 232.
- standard for serial comm unicaha that allow exchange of data btn computer and peripheral devices like printer or modem.

RS232

- Serial ~~standard~~ communicaha standard used for connecting computers and peripheral devices like modem.

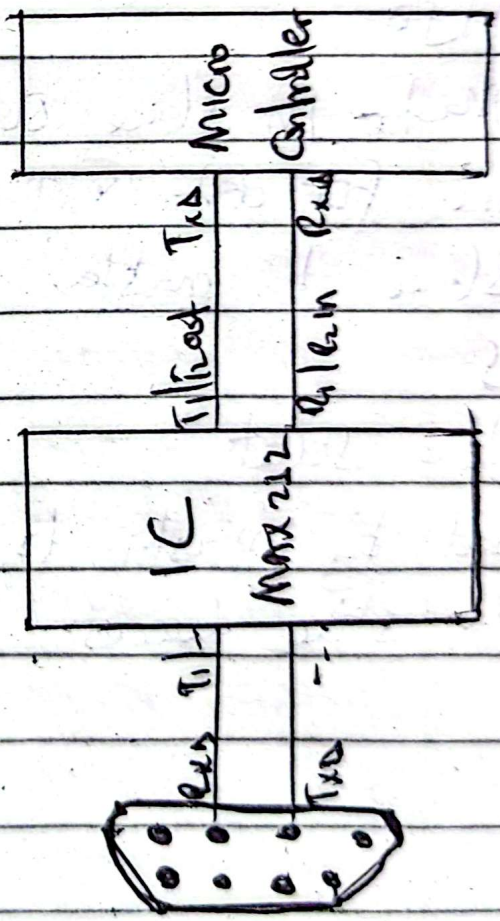
it has

1. start bit
2. Data bit (8bit).
3. ~~Sto~~ Parity bit
4. Stop bit

Usage

- Used in legacy systems, embedded system etc but replaced by USB, Ethernet wireless

Interfacing diagram [RS 232]



External Communication Interface USB.

USB - Universal Serial Bus

- It a standard for connecting communicating and supply power between computers and electronic devices like phones, printers etc.

Features

1. Plug and play
   - Devices are detected and configured automatically.
2. Hot-Swappable.
   - Can be connect / disconnect without restart system.
3. Power + Data
   - Provide data and power over the single cable

3 generations of USB specifications

- USB 1.x
- USB 2.0
- USB 3.x

Types of USB

- Min USB
- Micro USB
- USB Type-C
- Mini USB.
  - Divided into three types A type, B type and AB type
  - Older phones, and cameras.

Micro USB

- Reduced version of the USB
- connecting small & mobile devices

- including cameras, smartphones, GPS, MP3 players etc.

Types - Micro A, Micro B, Micro USB 2.

- Plug & play
- Hot - Swappable.

USB Type-C -

- Data, power are delivered to this type of USB
- Used in Smartphones & other USB - connected devices

Advantages of USB.

1. USB interface is self configuring, No setting on the devices
2. Hot pluggable & Play and play
3. Hot Swappable
4. Universal standard
   - one interface (Port) to many devices like keyboard, flash drives, phones etc.
   - Reduce multiple type of ports.
5. Power + Data on one cable.
6. High data speeds.

Disadvantages:

1. Limited cable length
   - Cables are short
2. Durability / Older types
   - Frequent plugging/unplugging can damage ports
3. Security Risk
   - Can easily spread malware

Qn. Compare RS-232 and USB communication standards in details.

- Both RS-232 and USB are used communication standards for connecting electronic devices and transferring data.

RS-232 - Recommended Standard 232
- It an older serial communication standard for short distance data exchange

USB - Universal Serial Bus
- Modern communication protocol used to connect multiple devices to computer with high data transfer speed.

Architecture & Working Principle

RS-232
- Uses Asynchronous Serial communication with no clock signal
- Data transmitted bit by bit over single channel
voltage levels
- Logic 1 = -3V to -15V
0 = +3V to +15V.

USB:

- Is synchronous protocol uses host controller architecture.
  - Support multiple devices
  - Support plug and play & hot-swapping
  - Transfer data in packets

Difference

Feature	RS-232	USB
1. Type	Serial [asynchronous]	Serial [synchronous]
2. Voltage level	±V3 to ±15V	0V to 5V [TTL]
3. Speed	Up to 115.2 kbps	Up to 4Gbps
4. Distance	up to 15 meters	5 meters.
5. No of devices	1 to 1 connection	up to 127 devices
6. Plug & Play	Not support	Support
7. Power supply	Not provided	Provides power to devices

Applications.

RS-232.

- Modems
- Old printers
- Industrial controllers
  - CNC Machines

USB.

- Keyboard, mice, printers.
- Flash drives, hard drives
- Cameras, mobile phones
- Microcontroller based development boards

UNIT V

ARM PROCESSOR.

Arm Architecture Version.

ARM - Stand for Advanced RISC Machine.

RISC - Reduced Instruction Set Computer.

ARM - Is a type of micro processor based on RISC architecture

1 - Is for commercial used

5. Operations are executed over register.
6. Many Thom processing Instruction use a 2-address format
6. Licensing Model
- Doesn't manufacture chip it license its design to companies

ARM Features.

- Arm is a 32-bit architecture
- Fixed length 32 bit instruction.
- All instruction can be executed conditionally

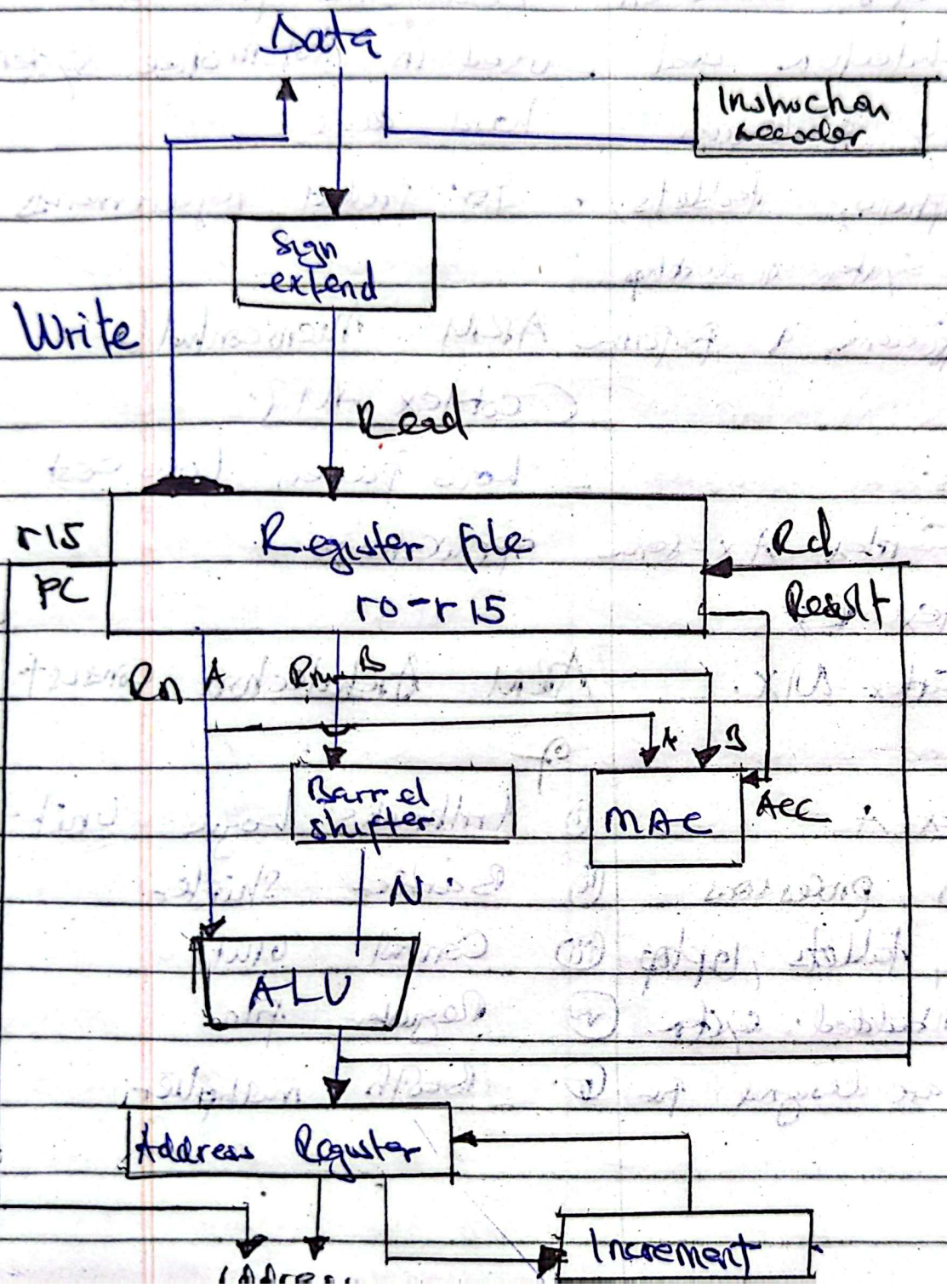
Implementation two instruction set
i) 32-bit ARM instruction
ii) 16-bit Thumb instruction

ARM Architecture

- It's Advanced Reduced Instruction set computing Machine (RISC)
- It's 32 bit RISC microcontroller.

ARM Cortex.

ARM ARCHITECTURE.



ARM cortex

- Family of CPU based on ARM architecture used on variety of devices like smartphone, tablets, embedded system or desktops
- Power efficiency & performa

Subfamilies.

1. ARM Cortex Ax-series
2. ARM Cortex Rx
3. ARM cortex Mx.

1. ARM Cortex A
- Application processors in smartphone, tablets, laptop & high embedded syste.
- This CPU are designed for complex

ARM Cortex R

- Real time processors.
- used in Automotive system hard disk
- For predict requirements

ARM Microcontroller Cortex -M3.

- low power, low cost applications

ARM Architecture consist of

(i) Arithmetic Logic Unit.
(ii) Barrier Shifter
(iii) Control unit
(iv) Register file.
(v) Booth multiplier.

ALU

- ALU has 32-bits input
- Primary come from register and other come from shifter.
- Status register flags modified by ALU output

Booth Multiplier

- Has 3 32-bit input & input return from register file.
- Multiplier output is barrel 32 bSB

Barrel shifter.

- Features 32-bit input to be shifted
- input coming back from register file. or might be immediate data.

Incremental

- For load and store instructions, incremental update the contents of address register before processor core reads or writes the next register.

Instruction decoder

- decode instruction opcode read from the memory & instruction executed

Register file

- 32 bit register used for storing data items

Pipeline.

Pipeline Concepts in ARM Processor.

- Pipelining
  - Concept used in ARM processors to improve instruction execution speed by overlapping different stages of instruction processing
  - It allows multiple instructions to be processed in parallel
  - Instruction divided into several stages, each performing specific tasks in instruction execution cycle.
  - Allow processor to work on multiple instructions at same time.

3-stage Pipeline in ARM:

1. Fetch
2. Decode.
3. Execute.

Fetch

- Instruction is fetched from memory
- It retrieve the next instruction to execute.

Decode.

- Fetched instruction is decoded to understand what is should do

Execute

- The instruction is executed and the result is written back if needed.

How it works.

- One instruction is decoded, next one is being fetched and previous one might be in executed stage.

Advantages

- Improve CPU performance & efficiency.
- Less time.
-

Advanced Pipelines.

- Have 5-stage or even deeper pipelines for further increase speed.

Ex ARM7, ARM9 etc.

Interrupt.

Software Interrupt in ARM.

- Instruction used in ARM processor to generate the software interrupt, allow the processor to switch to the supervisor mode & execute a specific service routine.

Functions

- SWI request a system level service
- The processor saves the current state
- Switches to supervisor mode.
- Jump to SWI vector address to execute the interrupt handler.

ARM Instruction Set.

- Each instruction is 32-bits long or 16-bits in Thumb mode
- Instruction are designed to execute in one clock cycle, make it very fast.

Instructions

1. Data Processing Instruction
2. Load & Store Instruction
3. Branch Instruction
4. Software Interrupt Instruction
5. Program Status Register Access
6. Multiply & Multiply-Accumulate Instruction
7. Condition of Execution

1. Data Processing Instruction
- These instruction perform ALU operations on registers.

Eg:

ADD R0, R1, R2
R0 = R1 + R2.

SUB R4, R5, #10
R4 = R5 - 10.

- used for addition, subtraction and logical operations like AND, OR, XOR etc

2. Load & Store Instruction.
- These instructions are used to move data b/w registers & memory.

Eg:

LDR R0, [R1]
- Load data from address R1 to R0

STR R2, [R1, #4]
- Store R2 to memory at address R1 + 4.

Uses:
- Accessing memory to read or write data.

3. Branch Instructions.
- used for changing the flow of execution or to control the flow. Use. Loop
jumps
Function Calls

Eg:

B Loop: Unconditional branch to Label Loop

BL Function
- Branch to subroutine function.

CPSR - Current Program Status Register
SPSR - Saved Program Status Register.

"BX LR - Return from subroutine using Link Register

Software Interrupt Instruction
- used to make a system call or trap to the OS

Eg:
SWI 0x01 - Request service with code 0x01

uses.
- used in os based system for user to kernel

Program Status Register Access:
- This instruction used to access or modify the CPSR and SPSR.
- used to control processor state

Multiply
- This instruction perform multiplication of register values

Eg
MUL R4, R2, R3.
R4 = R2 x R3

Conditional execution.
- Most ARM instructions can be conditionally executed based on the flag in CPSR
- It reduces need for branching.
- Improve pipeline performance

OMAP - PROCESSOR.

- stand for Open Multimedia Application Platform.
- Are designed for mobile phones, tablets, multimedia devices & embedded system.
- Built in on ARM architecture including ARM Core, DSP, and GPU.

GPU - Graphics Processing Unit
DSP - Digital Signal Processor

OMAP
- Are developed to handle multiple multimedia tasks like video, audio & gaming.
- Combine both ARM core with DSP and GPU.

Features & Function of OMAP Processor.
1. ARM Core
2. Heterogeneous Architecture
3. Multimedia support
4. Low Power Consumption
5. High Integration
6. Real-Time Operating System support

ARM Core
- uses ARM Cortex-A8, Cortex A9 or 15.
- support high-performance
- Low power consuming.

Heterogeneous Architecture,
- include combination of
  - ARM CPU - General processing
  - DSP
  - GPU

3. Multimedia support.
- Support 1080p HD video
- Hardware acceleration for audio, video & image processing

Low Power Consumption
- Designed used power optimization techniques like dynamic voltage for battery operated devices

High Integration.
- Contain memory controller, peripheral interfaces (USB, SPI, I2C and) and wireless communication (like Wi-Fi, Bluetooth) in single chip

Real-Time OS support
- Support OS like Android, Linux & Windows Embedded.

Register Bank in ARM Processor.
- used to store temporary data, control flags, return address & status information regarding data access

① General Purpose Register.
- Stack pointer
- Link Register
- Program Counter
  - Hold address of next
  - To store address after function.

Q: Program status Registers
- CPSR
- SPSR

Register Banking
- ARM architecture used support register banking where certain registers [SP, LR, SPSR] have multiple copies for different processor modes - User/System mode
- IRQ modes
- Speed up interrupt handling
- Fast data accesses & temporary storage
- Fast execution.
- Context switching

Conditional Flags used in the ARM Processor
- N - Negative flag.
- Z - Zero flag
- C - Carry flag
- V - overflow flag.

N - Negative flag
- Set to 1 if result of an operation in negative MSB is 1.

Z - Zero flag
- Set to 1 if result of an operation is zero
- used to check success full completion.

C - Carry flag
- Set if operation result is any out of borrow.

V - Overflow Flag

- Set when there is a signed overflow