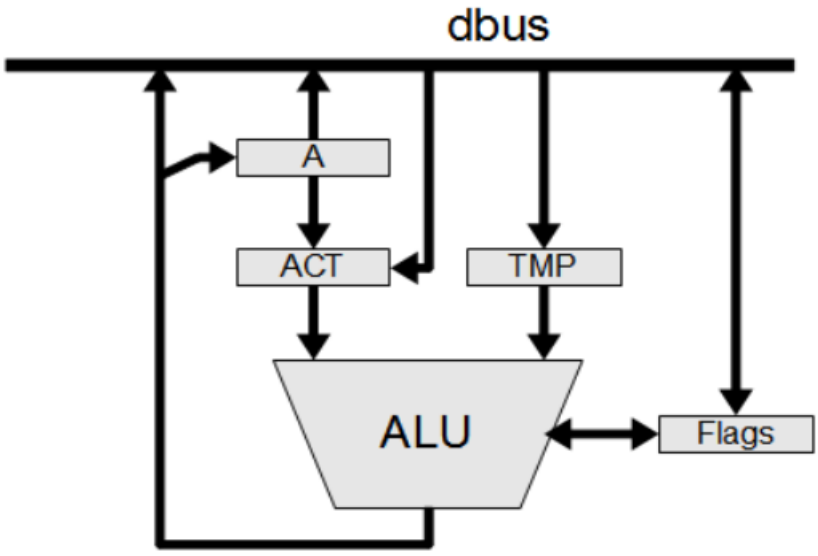
SEMESTER II – MICROPROCESSOR & MICROCONTROLLER EXAM FULL.
TOTAL MARKS: 50
DURATION: 2 HOURS

PART A (7 x 2 = 14 Marks)

1. What are the operations performed by ALU of 8085?
The Arithmetic Logic Unit (ALU) in 8085 performs the following operations:

- **Arithmetic Operations:** Addition, subtraction, increment, decrement.
- **Logical Operations:** AND, OR, XOR, NOT, compare.
- **Bitwise Operations:** Rotate (left/right), complement.
- **Flag Manipulation:** Updates status flags based on result (Zero, Sign, Carry, etc.).

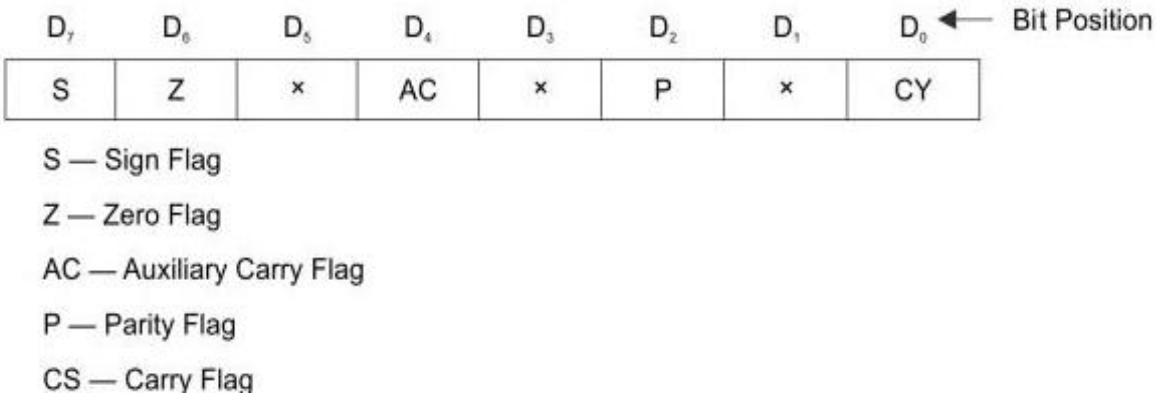


2. What are the functions of an accumulator?
The **accumulator** is an 8-bit register used in arithmetic and logic operations. It serves the following roles:

- Temporarily holds data to be processed.
- Stores the result of ALU operations.
- Acts as a source/destination register for data transfer.

3. Define flag register. (Reference ya mwalimu age 20)
The **flag register** is a special-purpose register that indicates the status of ALU operations. It consists of:

- **Sign Flag (S):** Set if result is negative.
- **Zero Flag (Z):** Set if result is zero.
- **Auxiliary Carry Flag (AC):** Used in BCD operations.
- **Parity Flag (P):** Set if result has even number of 1s.
- **Carry Flag (CY):** Set if there's a carry out from MSB.



4. List the segment registers of 8086. (Reference ya mwalimu 164)

Segment registers are 16-bit registers used to manage memory segments, allowing the processor to access a larger address space than it could with a simple 16-bit address.

8086 has **four segment registers**:

- **Code Segment (CS)** – Holds base address of instruction code.
- **Data Segment (DS)** – Points to data area.
- **Stack Segment (SS)** – Used for stack operations.
- **Extra Segment (ES)** – Additional data segment for string operations.

5. List the various string instructions available in 8086. (Reference age 237 Notes za Mwalimu**)**

String instructions falls into string **movement, comparison, scan, load and store.**

- `MOVS` – Move string.
- `LODS` – Load string to accumulator.
- `STOS` – Store string from accumulator.
- `SCAS` – Compare string.
- `CMPS` – Compare two strings.
- `REP` – Repeat prefix for loops.

6. What are the different ways of operand addressing in 8051? (Reference ya Mwalimu page 367)

Addressing mode are the ways used by CPU to fetch data.

- **Immediate Addressing:** Direct value given (e.g. `MOV A, #45H`).
- **Register Addressing:** Data in register (e.g. `MOV A, R0`).
- **Direct Addressing:** Access internal memory (e.g. `MOV A, 30H`).
- **Indirect Addressing:** Access via pointer register (e.g. `MOV A, @R0`).
- **Indexed Addressing:** Used with look-up tables (e.g. `MOVC A, @A+DPTR`).

7. What is data pointer (DPTR)?(Reference ya Mwalimu 312 **)**

The **DPTR** is a 16-bit register in 8051 used to hold memory addresses for external data memory. It is typically used in indexed addressing modes.

PART B (3 x 4 = 12 Marks)

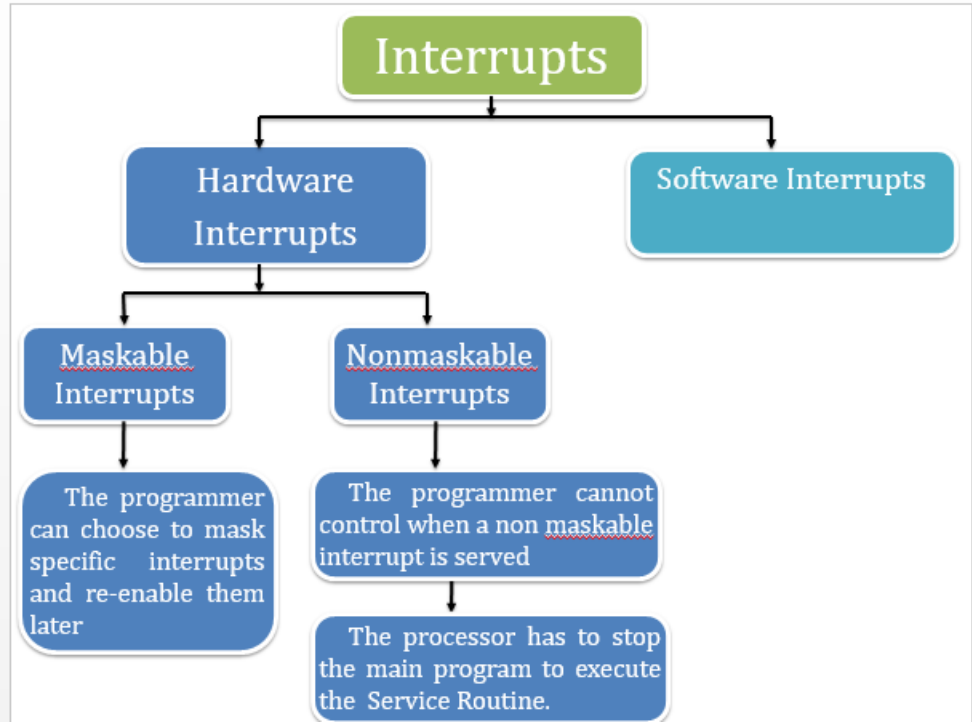
8(a). How many machine cycles does 8085 have, mention them. (Reference page 140 notes za mwalimu **)**

Machine cycle is The time required by the micro processor to complete the operation of accessing memory or I/O device.
8085 has **five types of machine cycles**:

- **Opcode Fetch Cycle** – Fetches instruction opcode.
- **Memory Read Cycle** – Reads data from memory.
- **Memory Write Cycle** – Writes data to memory.
- **I/O Read Cycle** – Reads data from I/O device.
- **I/O Write Cycle** – Writes data to I/O device.

OR

8(b). How many types of interrupts are there in 8085? Which interrupt has the highest priority? (Reference page 123 **)**



8085 has **5 hardware interrupts**: (Reference age ya 126)

- **TRAP** – Highest priority, non-maskable.
- **RST7.5**
- **RST6.5**
- **RST5.5**
- **INTR** – Lowest priority.

TRAP has the highest priority.

In 8085, hardware interrupts, specifically the TRAP interrupt, generally have higher priority than software interrupts

9(a). Define Macros. Give an example. (Reference ya mwalimu page 266 **)**

A **Macro** is a sequence of instructions/or procedures(but note really) given a name and expanded inline during assembly.

- Its syntax
- Example (Macro definition)
- **name** MACRO [parameters,...]
- <instructions>
- ENDM
-

Example:

```
MYMACRO MACRO
    MOV A, #55H
    MOV B, A
ENDM
```

OR

9(b). What is an assembler directive? (Reference ya mwalimu page 256)

Assembler directives are instructions to the assembler, not the CPU. They control code organization, memory, and macros.

Also known as pseudo-ops, are instructions to the assembler that guide the assembly process.

- **Used to :**

› specify the start and end of a program

› attach value to variables

› allocate storage locations to input/ output data

› define start and end of segments, procedures, macros etc.

Examples: ORG, DB, EQU, END, SEGMENT

10(a). What is indexed addressing? (reference ya mwalimu Page 206)

In **indexed addressing**, refer to the addressing mode where the effective address is obtained by adding a base address (like DPTR or PC) to an offset (like A).

Example: `MOVC A, @A + DPTR`

OR

10(b). Compare Microprocessor and Microcontroller. (reference ya mwalimu)

DIFFERENTIATING FEATURES

FEATURES	MICROPROCESSOR	MICROCONTROLLER
Function	**Process** the **general** task, only	Both **Process** and **Control** the **specific** task
Memory	**No** in-built memory	In-built ROM and RAM memories
Application	**General** purpose (eg. PC, modems, printers)	**Specific** purpose (eg. A.C. and washing machine, home automation)
Complexity	**More** complex, **large** no. of instructions	**Less** complex, **less** no. of instructions
Cost	**High** (design time is more)	**Low** (design time is low)
Efficiency	**Less**	**More**
Architecture	**Von Neumann** (program and data stored in same memory)	**Harvard** (program and data stored in different memory)
Example	8085, 8086	8051

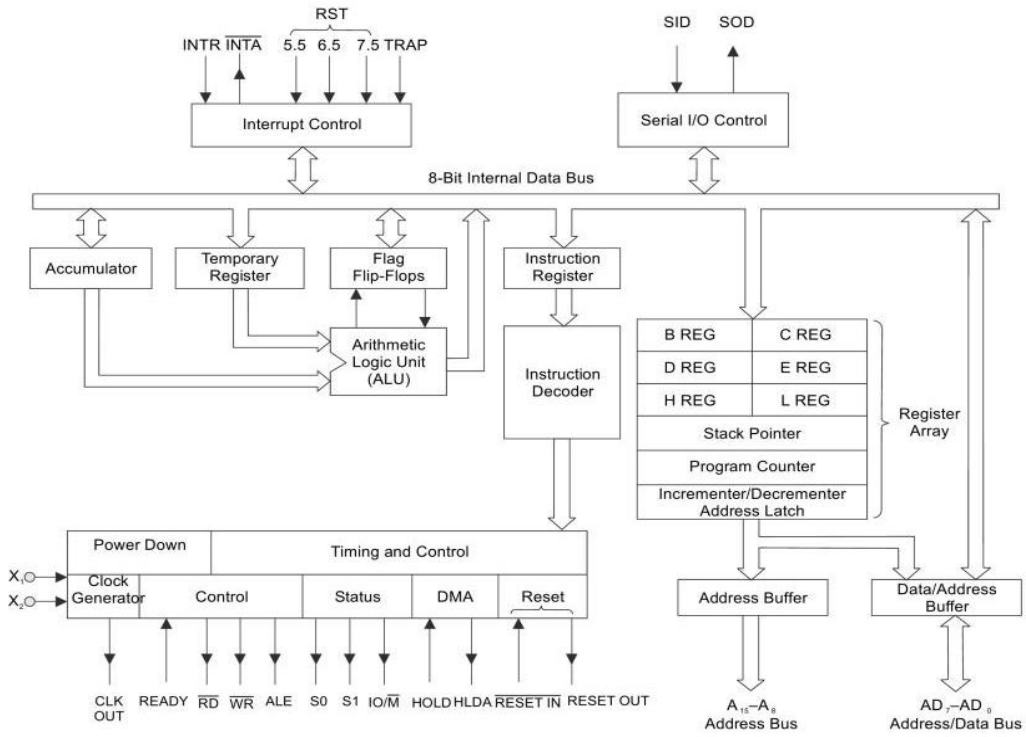
Summary additions:

Feature	Microprocessor	Microcontroller
Function	CPU-only	CPU + Memory + I/O
Memory	External	Built-in
Application	General-purpose computing	Embedded systems
Power	High	Low

PART C (2 x 12 = 24 Marks)

11(a). Explain the architecture of 8085 microprocessor. (reference ya mwalimu page 8)

8085 is an 8-bit microprocessor with the following components:



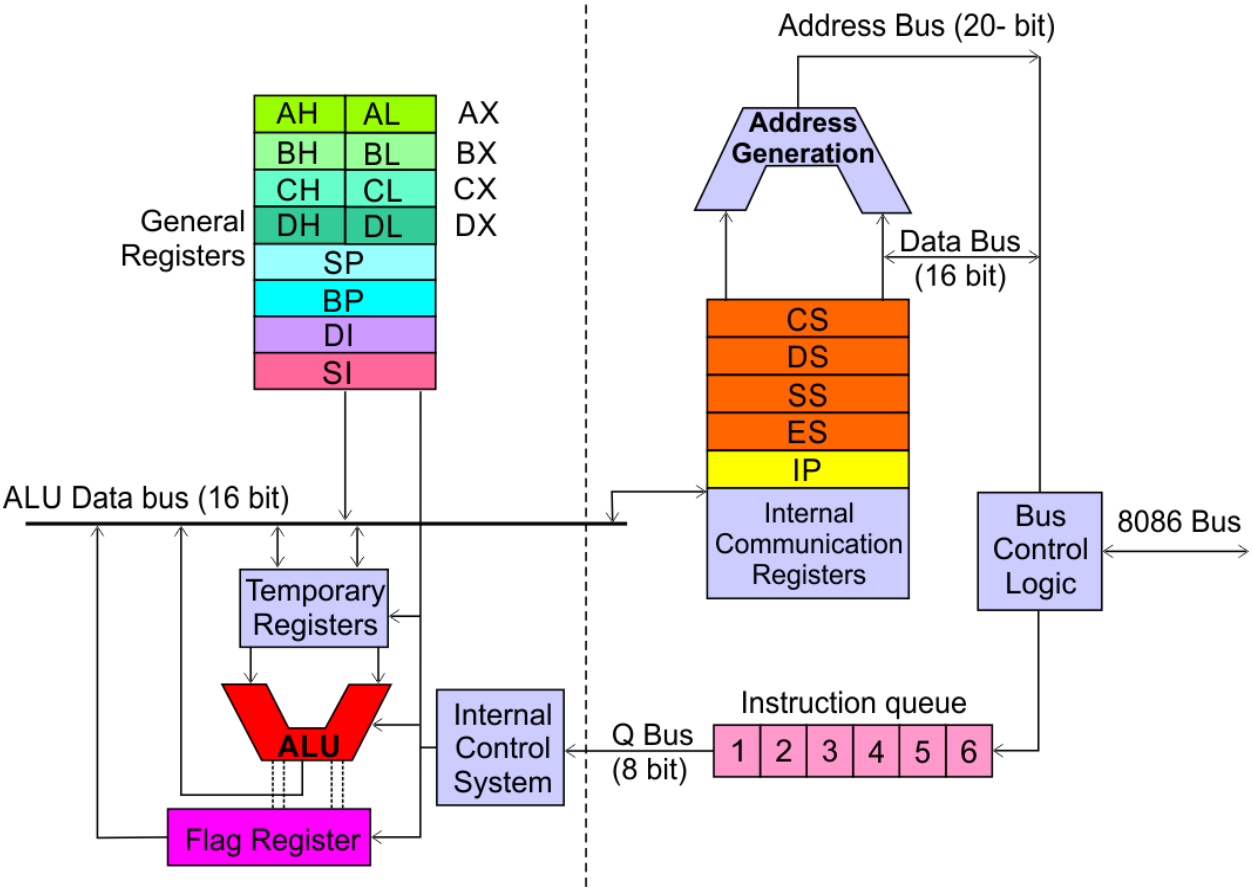
- **ALU:** Performs arithmetic/logical operations.
- **Accumulator:** Stores ALU results.
- **Registers:** B, C, D, E, H, L for temporary data.
- **Program Counter (PC):** Points to next instruction.
- **Stack Pointer (SP):** Points to top of the stack.
- **Instruction Register & Decoder:** Decodes instructions.
- **Timing & Control Unit:** Manages all control signals.
- **Address & Data Bus:** 16-bit address, 8-bit data.

OR

11(b). Explain the types of instructions in 8085 with example. (reference ya mwalimu page 62 **)**

- **Data Transfer Instructions:** `MOV A, B`
- **Arithmetic Instructions:** `ADD B`, `SUB C`
- **Logical Instructions:** `ANA B`, `XRA C`
- **Branching Instructions:** `JMP 2000H`, `CALL 2450H`
- **Control Instructions:** `NOP`, `HLT`, `DI`, `EI`

12(a). Explain internal hardware architecture of 8086 with neat diagram. (reference page 162)



8086 is a 16-bit microprocessor with:

- **EU (Execution Unit):** Executes instructions using ALU, register array, flag register.
- **BIU (Bus Interface Unit):** Handles address/data buses, prefetch queue.
- **Prefetch Queue (6 bytes):** Improves speed via pipelining. (used for implementation of pipelining)
- **Segment Registers:** CS, DS, SS, ES.

OR

12(b). Explain the types of instructions in 8051 with example. (reference ya mwalimu 338 **)**

- **Data Transfer:** `MOV A, R1`
  - **PUSH / POP**
    - Push and Pop a data byte onto the stack.
  - **MOV destination, source** ; copy source to destination.

- **Arithmetic:** `ADD A, #10H`
  - **ADD A, source** ;ADD the source operand to the accumulator
    - MOV A, #03H ;load 03H into **A**
      MOV B,#02H ;load 02H into **B**
      ADD A,B

- **Logical:** `ANL A, #0F0H`
  - **ANL D,S**
    - -Performs **logical AND** of destination & source
    - -Destination : A/memory;

    -Source : data/register/memory
    - - Eg: **ANL A,#0FH ANL A,R5**
  - **ORL D,S**
    - -Performs **logical OR** of destination & source
    - -Destination : A/memory;

    -Source : data/register/memory
    - - Eg: **ORL A,#28H ORL A,@R0**

- **Branching:** `SJMP LABEL`, `CJNE A, #20H, LABEL`
  - Program branching instructions are used to control the flow of actions in a program
  - Some instructions provide decision making capabilities and transfer control to other parts of the program.

- **Bit Manipulation(Boolean):** `SETB P1.0`, `CLR C`
  - **CLR:**
    - The operation **clears** the specified bit indicated in the instruction
    - Ex: CLR C clear the carry
  - **SETB:**
    - The operation **sets** the specified bit to **1**.
  - CPL:

- The operation **complements** the specified bit indicated in the instruction

---

TEST II – ADVANCED MICROPROCESSOR & MICROCONTROLLER MODEL ANSWERS

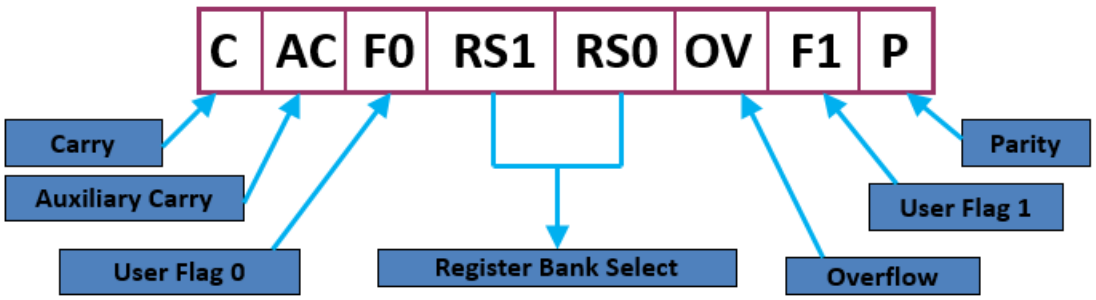
PART A (7 x 2 = 14 Marks)

1. Mention the function of the PSW register in 8051.

The Program Status Word (PSW) register contains status bits like

- Carry,
- Auxiliary Carry,
- Register Bank Select bits,
- Overflow, and Parity,

Which help in decision-making during program execution.



2. How many ports are there in 8051 and what are their names? (reference notes za mwalimu page 312 also in the diagram, ALSO Page 420 **)**

There are 4 parallel I/O ports: **P0, P1, P2, and P3**. Each port is 8-bit wide and can be used for input or output.

3. Differentiate between ADC and DAC. (reference ya mwalimu 512)

- **ADC (Analog to Digital Converter):** Converts analog signals to digital.
- **DAC (Digital to Analog Converter):** Converts digital signals to analog.

4. What is PC Bus?

PC Bus refers to the communication pathway used to connect peripheral devices to the CPU in personal computers, often adhering to ISA or PCI standards.

5. What is pipelining in an ARM processor? Mention its advantage. (reference ya mwalimu 575)

Pipelining is a technique where multiple instruction phases are overlapped.

Basic 3-Stage Pipeline in ARM:

- **Fetch** – The instruction is fetched from memory.
- **Decode** – The fetched instruction is decoded to understand what it should do.

- **Execute** – The instruction is executed, and the result is written back if needed

Advantage:

- Increases instruction throughput and
- Improves overall speed.
- Improves CPU performance and efficiency.
- Makes better use of processor resources.

6. What is the function of the Program Counter (PC) in an ARM processor? (reference ya mwalimu page 612 **)**

- The PC holds the address of the next instruction to be executed.
- It helps the processor keep track of program sequence.

7. List any two features of ARM architecture.

- **RISC-based design** with fewer instructions.
- **Low power consumption** suitable for embedded/mobile devices.

PART B (3 x 4 = 12 Marks)

8(a). What is DAC? Why is it important to interface a DAC with a microcontroller in embedded systems?

A DAC converts digital values into analog signals. It's important for:

- Controlling analog devices (motors, speakers).
- Generating audio/video output.
- Signal modulation in communication systems.

OR

8(b). List any eight features of the 8051 microcontroller. (reference ya mwalimu page 291 **)**

1. 8-bit CPU
2. 4 I/O ports
3. 128 bytes RAM
4. 4K ROM
5. 2 Timers/Counters
6. Serial Communication Control
7. Interrupt system (*6-source/5-vector interrupt structure with two priority levels*)
8. Power-saving modes (Idle and Power-down)

9(a). Explain the SPI bus used in microprocessor systems. (reference ya mwalimu page 540)

SPI (Serial Peripheral Interface) is a synchronous, full-duplex communication protocol used for high-speed data transfer between microcontrollers(master) and peripherals(slaves).

- Uses 4 lines: MISO, MOSI, SCLK, SS
- Fast, simple, and efficient
- Suitable for sensors, memory, ADCs, etc.

b) Four main signal lines in SPI:

- SCLK (Serial Clock): Clock signal generated by the 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 the master to select the slave device for communication.

c) Two advantages of SPI over I²C:

- Faster data transfer speeds due to full-duplex communication.
- Simple hardware implementation without addressing, reducing complexity

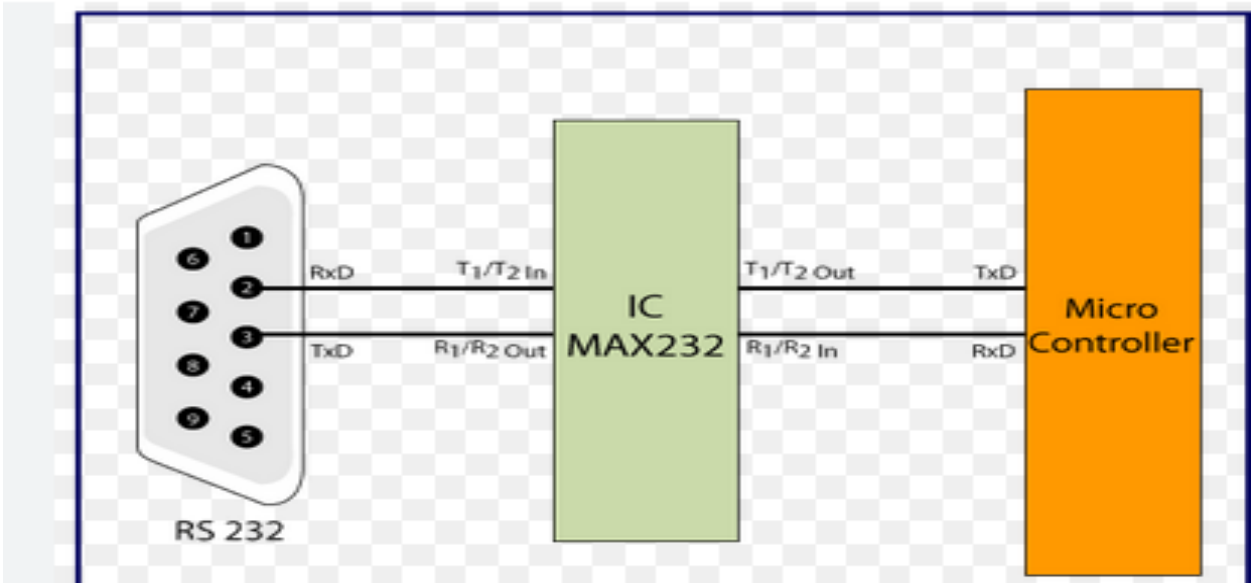
OR

9(b). How the RS-232 serial bus is interfaced to 8051 microcontroller? (Reference ni page 544 **)**

RS-232 is interfaced to 8051 via

UART (Universal Asynchronous Receiver Transmitter) and a

level shifter (e.g., MAX232) to convert voltage levels. TX and RX pins are connected to 8051 P3.0 and P3.1.



Mfano wa Level Shifter MAX232

10(a). What is the Thumb instruction set in ARM processors? Mention its advantages. (reference ya mwalimu ni 617 **)**

Thumb is a 16-bit compressed instruction set of ARM that helps to

- Saves memory
- Improves code density

Advantages of Thumb Instructions:

- Reduces code size – Efficient for embedded systems with limited memory.
- Improves performance – More instructions can be fetched per memory access.
- Saves power – Smaller code results in lower power consumption.
- Compatibility – ARM cores can switch between ARM and Thumb modes.

OR:

10(b). List and explain the different conditional flags used in the ARM processor. (reference ya mwalimu ni page 616)

- **N (Negative):** Set if result is negative
- **Z (Zero):** Set if result is zero
- **C (Carry):** Set if carry out occurs
- **V (Overflow):** Set if signed overflow occurs

PART C (2 x 12 = 24 Marks)

11(a). Explain the various addressing modes supported by the 8051 microcontroller. (reference ya mwalimu page 367 **)**

- **Immediate:** `MOV A, #25H`
- **Register:** `MOV A, R1`
- **Direct:** `MOV A, 30H`
- **Indirect:** `MOV A, @R0`
- **External direct (Termed as indexed in the table):** `MOVC A, @A+DPTR`

Addressing Modes in 8051

Addressing Mode	Syntax Example	Description	Where Data is Fetched From
Immediate	MOV A, #25H	Loads a constant value directly into a register.	Embedded in the instruction itself
Register	MOV A, R1	Transfers data between accumulator and general-purpose register,	From one of R0–R7 in selected register bank
Direct	MOV A, 30H	Accesses data stored in a specific internal RAM or SFR address	From internal RAM address 00H–7FH or SFRs 80H–FFH
Indirect	MOVC A, @R0	Accesses internal RAM indirectly through R0 or R1 (used as a pointer)	Address contained in R0 or R1
Indexed	MOVC A, @A+DTR	Used mainly to access code memory for lookup tables with an offset	Adds accumulator value to DPTR, and accesses code space,

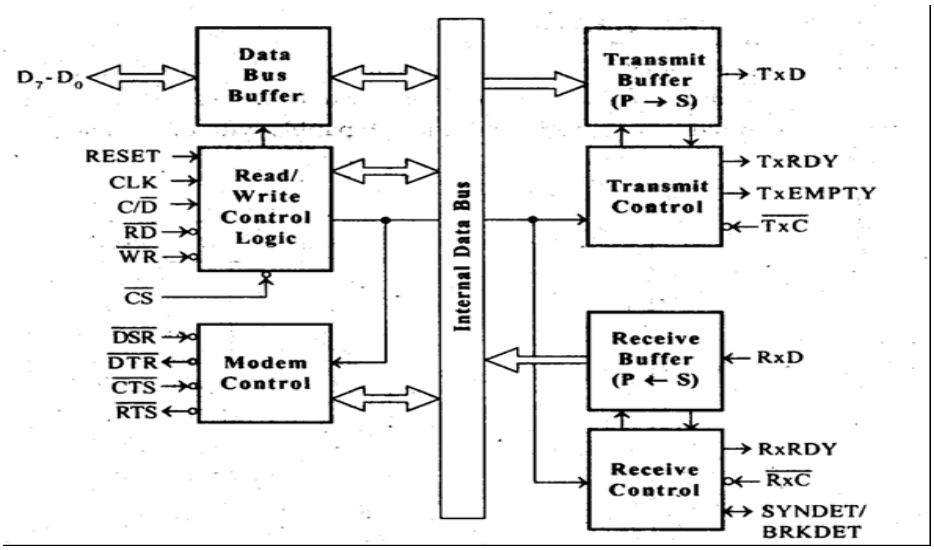
OR

11(b). Explain how the 8251 USART serves as an interface device between CPU and serial channels. (reference ya mwalimu 449)

acts as a mediator between microprocessor and peripheral to transmit serial data into parallel form and vice versa.

It takes data serially from peripheral (outside devices) and converts into parallel data. After converting the data into parallel form, it transmits it to the CPU.

Similarly, it receives parallel data from microprocessor and converts it into serial form. After converting data into serial form, it transmits it to outside device (peripheral).



8251 converts parallel data from CPU into serial form and vice versa. Features:

- Supports synchronous/asynchronous modes
- Provides status registers
- Manages baud rate, parity, stop bits
- Interfaces via control and data bus

12(a). Compare RS-232 and USB communication standards. Working principle, architecture, advantages, disadvantages and appplications. (Reference ya mwalimu 5.)

Feature	RS-232	USB
Type	Serial (asynchronous)	Serial (synchronous, packet-based)
Voltage Levels	±3V to ±15V	0V to 5V (TTL)
Speed	Up to 115.2 kbps	Up to 10 Gbps (USB 3.1/3.2)
Distance	Up to 15 meters	Up to 5 meters (standard USB)
Number of Devices	1-to-1 connection	Supports up to 127 devices
Connector Type	DB9, DB25	USB-A, USB-B, USB-C
Plug and Play	Not supported	Supported
Power Supply	Not provided	Provides power to devices

RS-232:
RS-232 uses **asynchronous serial communication** with no clock signal.
Data is transmitted **bit by bit** over a single channel using **start and stop bits**.
Uses **voltage levels**:
Logic 1 = –3V to –15V
Logic 0 = +3V to +15V
Uses **DB9 or DB25 connectors**.

USB:
USB is a **synchronous protocol** that uses a **host-controller architecture**.
It supports **multiple devices** via hubs using a **tiered star topology**.
Uses **differential signaling** (D+ and D– lines).
Supports **plug-and-play** and **hot-swapping**.
Transfers data in **packets** with built-in **error checking and retry mechanisms**.

RS-232:

Advantages:

- **Simple and easy to implement.**
- **Good for short-distance point-to-point communication.**

Disadvantages:

- **Slow data transfer rate.**
- **Limited to one device.**
- **Large and bulky connectors.**

USB:

Advantages:

- **High-speed data transfer.**
- **Supports multiple devices.**
- **Provides power supply to devices.**
- **Easy to use with plug-and-play.**

Disadvantages:

- **More complex protocol.**
- **Requires driver support in the operating system.**

RS-232 Applications:

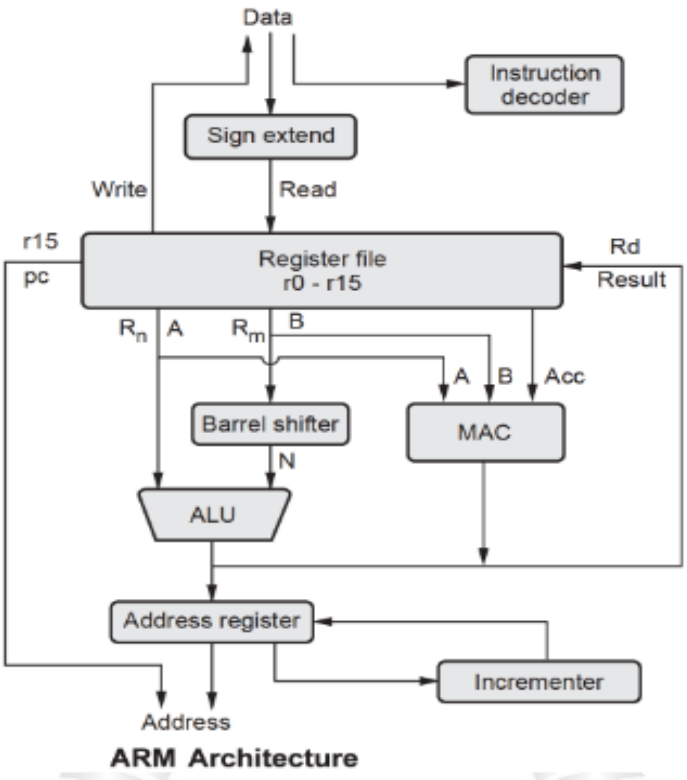
- **Modems**
- **Old printers**
- **CNC machines**
- **Industrial controllers**

USB Applications:

- **Keyboards, mice, printers**
- **Flash drives, hard drives**
- **Cameras, mobile phones**
- **Microcontroller-based development boards**

OR

12(b). Explain in detail the architecture of ARM Processor. (Reference ya mwalimu ya 571)



ARM Architecture

- **ARM = Advanced RISC Machine**
  - Key Features: RISC architecture, 32-bit or 64-bit, low power
  - Components: ALU, barrel shifter, register file, instruction decoder, pipeline stages
  - Memory support: Harvard or Von Neumann
  - Application: Mobile, embedded, real-time systems

- The ARM architecture processor is an advanced reduced instruction set computing [RISC] machine and it's a 32 bit RISC microcontroller.

- The ARM cortex is a complicated microcontroller within the ARM family that has ARMv7 design. There are 3 subfamilies within the ARM cortex family:
  - a) ARM Cortex Ax-series
  - b) ARM Cortex Rx-series
  - c) ARM Cortex Mx-series

• The ARM Architecture consists of following:
  - a) Arithmetic Logic Unit
  - b) Booth multiplier
  - c) Barrel shifter
  - d) Control unit
  - e) Register file

  - **a) Arithmetic Logic Unit**
    → Performs math and logic operations.
  - **b) Booth Multiplier**
    → Fast binary multiplication unit.
  - **c) Barrel Shifter**
    → Shifts bits in one cycle.
  - **d) Control Unit**
    → Directs instruction execution flow.
  - **e) Register File**
    → Holds temporary processor data.
