Assignment - 3.

Q) Write about stacks and subroutine.

A. Stacks and subroutines are fundamental concepts of computer science and programming. Here's an in-depth look at each:

Stacks

Definition

A stack is a data structure that follows the last in, first out (LIFO) principle. This means the last element added to the stack is the first one to be removed.

Operations

1. Push : Adds an element to the top of the stack.
2. Pop : Removes and returns the top element of the stack.
3. Peek/ Top : Returns the top element without removing it.
4. IsEmpty : Checks if the stack is empty.
5. Size : Returns the number of elements in the stack.

MJCET
• Return
out

Implementation

Stacks can be implemented using arrays & linked lists. In an array-based implementation, a fixed-size array and a variable (top) indicating the current position are used. In a linked list implementation, each node points to the next node, and a pointer to the top node is maintained.

Applications

- Function call management
- Expression evaluation.
- Undo Mechanisms
- Depth-First Search.

Subroutines

Definition.

A subroutine (also known as a function, method, or procedure) is a set of instructions designed to perform a specific task. Subroutines allow code reuse and modular programming.

Structure

A subroutine typically includes:

- Name: Identifies the subroutine.
- Parameters: Inputs to the subroutine.
- Body: The set of instructions to execute.

- Return type : The type of value the subroutine returns (if any).

Calling and execution

When a subroutine is called :

1. The calling code (caller) passes control to the subroutine.
2. The subroutine executes its instructions.
3. The subroutine may return a value to the caller.
4. Control returns to the caller, often to the statement immediately following the call.

Parameters.

- Pass-by-value : The subroutine receives a copy of the argument. Changes to the parameter do not affect the original argument.
- Pass-by-reference : The subroutine receives a reference to the argument. Changes to the parameters affect the original argument.

Both stacks and subroutines are essential for efficient and organized programming, enabling code reuse, modularity.

b) Write about interrupts in detail?

A. Interrupts are a fundamental concept in c
systems, providing a mechanism for the c
to respond to asynchronous events. They a
the processor to be alerted to important events
(such as input/output operations, errors, or other
critical conditions) and to handle these events
immediately, improving the efficiency and
responsiveness of the system. Here's a detailed
explanation of interrupts:

Definition.

An interrupt is a signal sent to the CPU by
hardware or software indicating an event that
needs immediate attention. When an interrupt
occurs, the CPU temporarily halts its current
operations, saves its state, and executes a
function known as an interrupt handler (or
interrupt service routine, ISR) to deal with the
event. After the ISR is executed, the CPU re-
stores its previous state and resumes normal
operations.

Types of Interrupts.

1. Hardware interrupts

- Maskable Interrupts (IRQs): These can be
enabled or disabled by the CPU using an

They are used for routine I/O operations.

- Non-Maskable Interrupts (NMIs):
These cannot be disabled and are used for critical events like hardware failures.
- Inter-Processor Interrupts (IPIs):
Used in multi-core systems to communicate between processors.

2. Software Interrupts:

- System Calls: Triggered by software applications to request service from operating systems.
- Exceptions: Triggered by errors or unusual conditions during program execution (e.g division by zero, invalid memory access).

Interrupt handling

1. Interrupt Request (IRQ) line: A hardware line over which a device can signal an interrupt to the CPU.

2. Interrupt Vector table (IVT):
A table of pointers to ISRs, where each entry corresponds to a specific interrupt.

3. Interrupt acknowledge : The CPU ... the interrupt request and prepares ... the interrupt.

JCET

at the ... ture, op... illustrat...

Conclusion

Interrupts are a fundamental aspect of computer architecture that enable efficient and responsive interaction between the CPU and external devices. or software events. By allowing the CPU to handle critical tasks promptly and manage multiple tasks effectively, interrupts play a vital role in the performance and functionality of modern computing systems.

4) Write about programmable interrupt controller (8259A) with necessary diagram?

A. The 8259A programmable Interrupt Controller (PIC) is a crucial component used in microprocessor-based systems to manage and prioritize multiple interrupt requests. The 8259A PIC is designed to allow a single processor to handle multiple interrupt sources efficiently. Here's a detailed look at

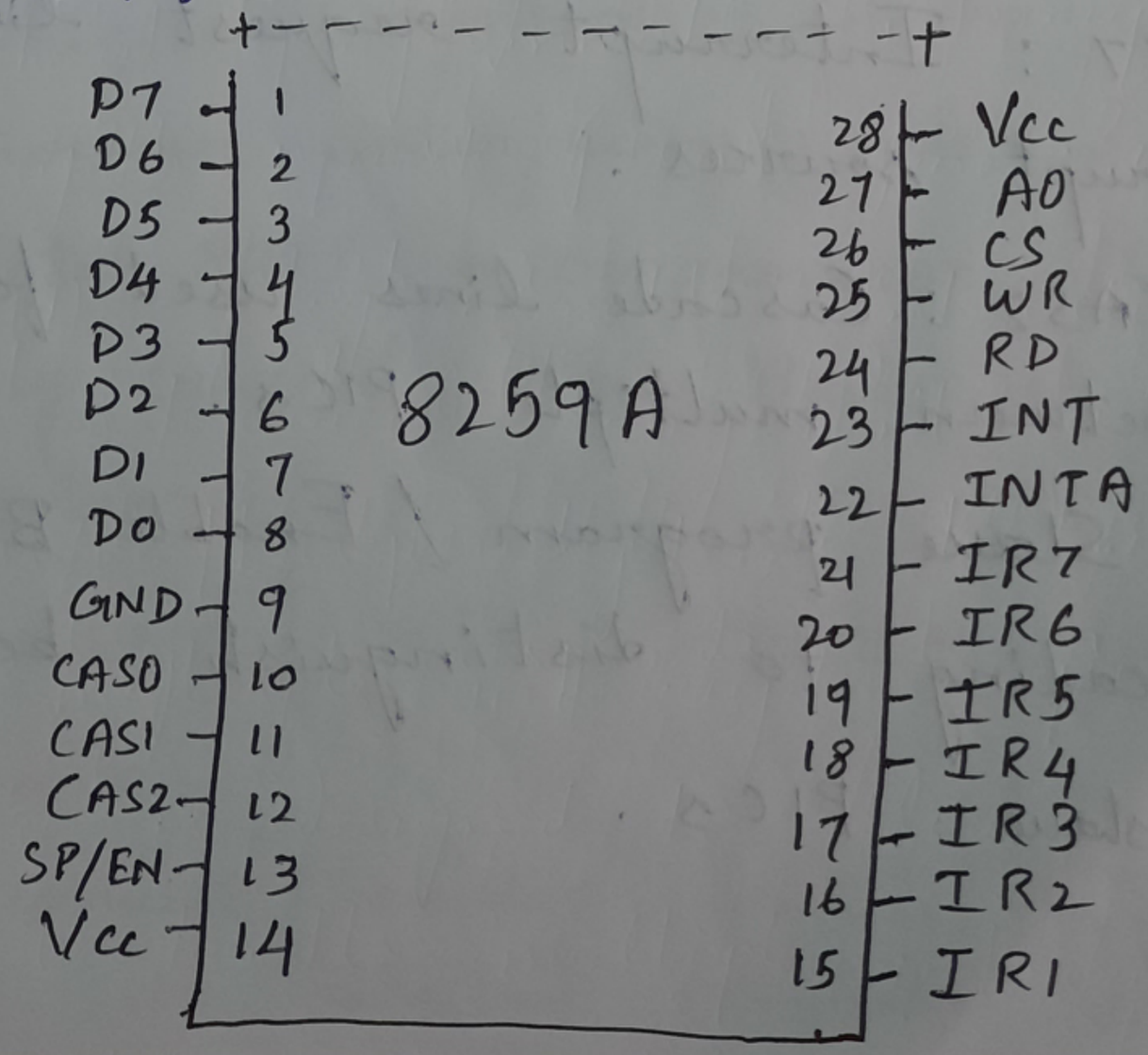
at the 8259A, including its architecture, operation and a diagram to illustrate its components.

Overview

The 8259A is an 8-bit device that can handle up to 8 interrupt requests (IR0 to IR7). It can be cascaded with other 8259A chips to handle up to 64 interrupt requests. The 8259A communicates with the CPU using an 8-bit data bus and control lines.

Architecture and Pin Diagram

Below is a simplified pin diagram of the 8259A:



Key pins and Signals

- D0 - D7 : Data bus lines used for data transfer between the PIC and the CPU.
- A0 : Address line to select control registers.
- CS : Chip Select, enables the 8259A for communication.
- WR : Write, indicates data on the data bus should be written on PIC.
- RD : Read, indicates data should be read from the PIC.
- INT : Interrupt request, the line through which the PIC sends an interrupt to the CPU.
- INTA : Interrupt Acknowledge, used by the CPU to acknowledge the interrupt.
- IR0 - IR7 : Interrupt request lines for up to 8 interrupt sources.
- CAS0 - CAS2 : Cascade lines used for communication between multiple PICs.
- SP/EN : Slave program / Enable Buffer, used in cascading to distinguish between master and slave PICs.

MJCET

Conclusio

The 8259

Conclusion.

The 8259A programmable Interrupt controll
→ is a vital component in managing multiple Interrupt requests in microprocess-or-based systems. Its ability to prioritize and handle interrupts efficiently makes it essential for ensuring responsive and stable operating system.

4] Write in detail about Analog to Digital and Digital to Analog converters with interfacing 8085.

A. Analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) are essential components in interfacing analog signals with digital systems. In the context of microprocessor interfacing, such as with the Intel 8085 microprocessor, these converters play crucial roles. Here's a detailed explanation:

Analog-to-Digital converters (ADCs)

1. Function:

ADCs convert analog signals (continuous signals) into digital signals (discrete

binary value). This conversion is necessary microprocessors can only process digital

2. Working principle:

- Sampling: The analog signal is sampled at regular intervals.
- Quantization: Each sample is assigned a finite set of discrete values.
- Encoding: These discrete values are then encoded into binary form.

3. Types of ADCs:

- Successive Approximation ADC: Uses a comparator and a digital-to-analog converter internally to converge on the input voltage step-by-step.
- Flash ADC: Uses a bank of comparators to convert the signal in a single step, providing very high speed conversion.
- Sigma-Delta ADC: Utilizes oversampling and noise shaping for high-resolution conversion, often used in audio applications.

4. Interfacing ADC with 8085:

- Selection of ADC: Common ADCs include ADC 0808/0809, which have an 8-channel multiplexer and an 8-bit output.

MJCET

- Contro
de

- Control Signals : Necessary signals include Chip Select (CS), Read (RD), Write (WR), and Start of Conversion (SOC).

Digital-to-Analog Converters (DACs):

1. Function :

DACs convert digital signals (binary values) into analog signals (continuous signals). This conversion is necessary when a digital system needs to control or generate analog signals.

2. Working principle :

- Binary weighted DAC : Uses resistors of different values weighted according to binary progression to generate the output voltage.
- R-2R Ladder DAC : Utilizes a repeating pattern of two resistors values (R and 2R) for simplicity, and precision.

3. Interfacing DAC with 8085 :

- Selection of DAC: Common DACs include DAC 0800, which has an 8-bit digital input and an analog output.

- Control signals : Typically include Chip (CS) and write (WR).
- Address Decoding : Proper address decoding necessary for ensuring the DAC responds to the correct addresses on the 8085 bus.
- Data transfer : The 8085 writes digital data to the DAC's input register to get the corresponding analog output.

By understanding these principles and how to interface ADCs and DACs with the 8085, one can effectively integrate analog and digital components in various applications such as sensor data acquisition, signal processing and control systems.

e) Explain about DMA Controller (Intel 8257) in detail with block and pin diagram.

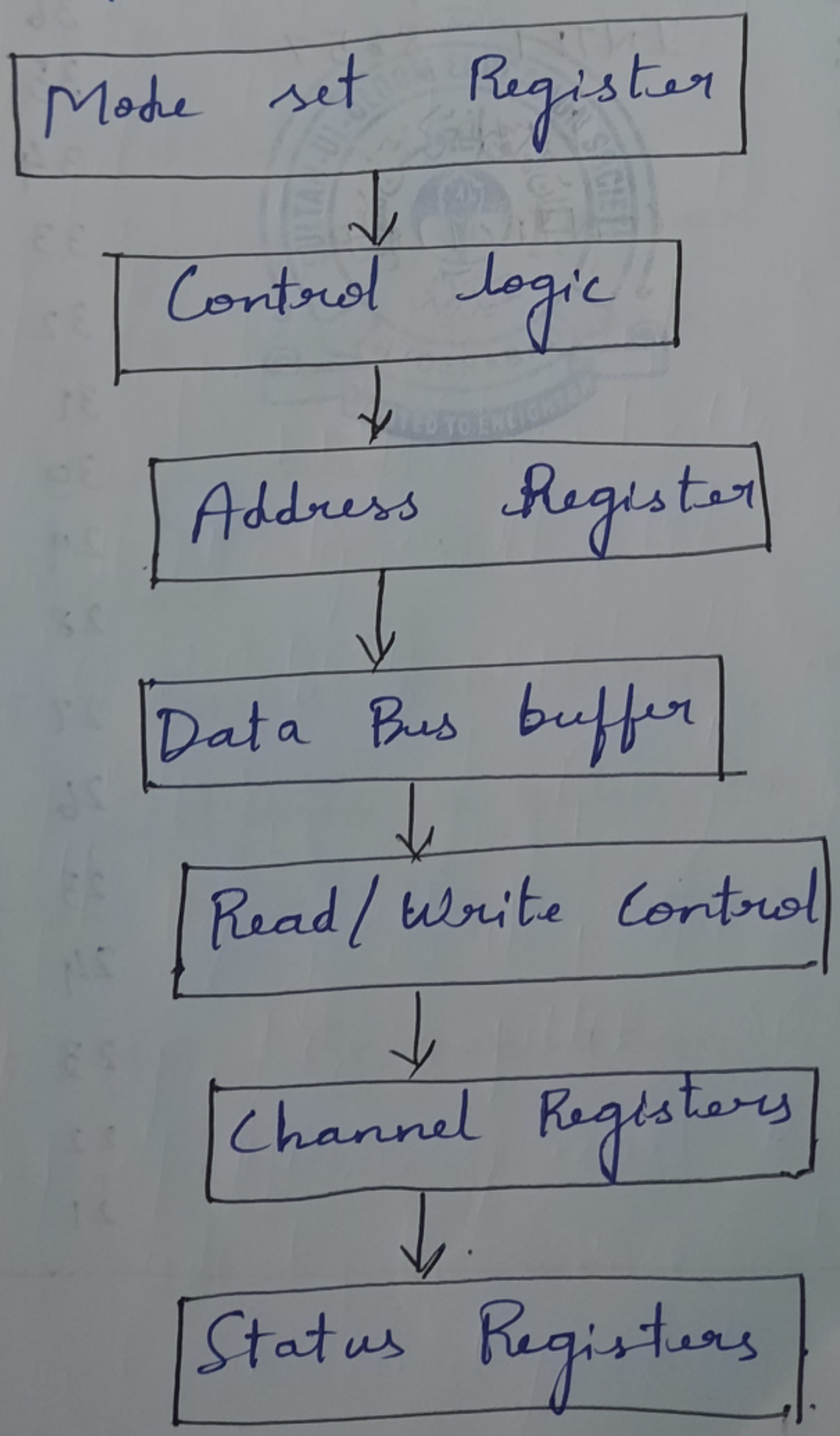
A. Intel 8257 DMA Controller.

The intel 8257 is a direct Memory Access (DMA) controller designed to transfer data directly between memory and peripherals without the continuous intervention of the CPU. This increases data transfer efficiency and speeds up the system's overall performance.

Key Features :

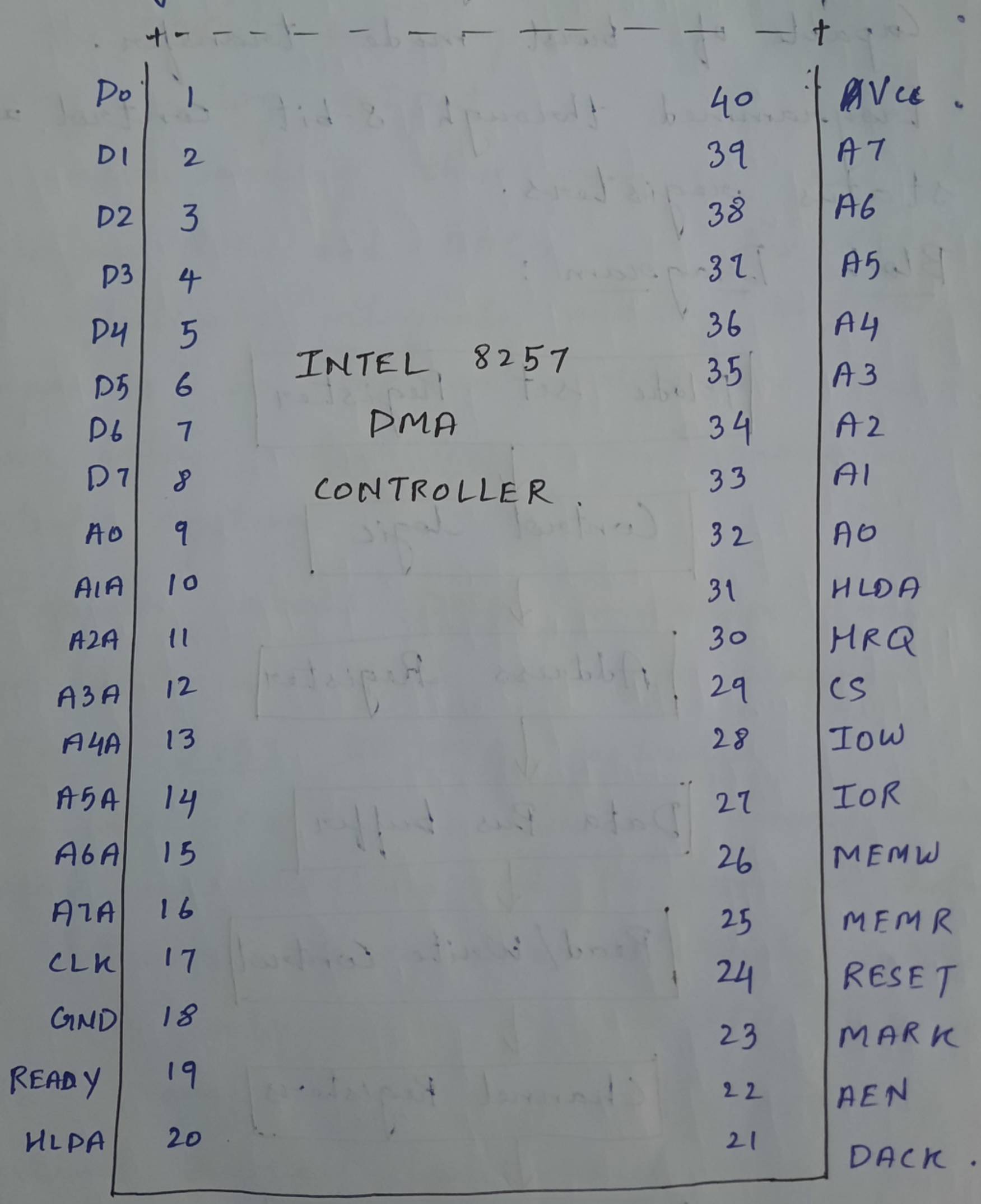
- Four independent DMA channels.
- Each channel can address 64 kB of memory.
- Supports memory-to-memory, I/O-to-memory, and memory-to-I/O data transfers.
- Capable of burst mode transfer.
- Programmed through 8-bit control and status registers.

Block Diagram:



The block diagram of the intel 8251 is co... of several funtional blocks, including the ad... Register, Control logic, Data Bus Buffer, Read/ Write Control logic, and Mode set Register.

Pin Diagram



Pin descriptions

- D0 - D7 :- Data Bus.
- A0 - A7 :- Address lines.
- AEN : Address enable.
- HRQ : Hold Request.
- HLDA : Hold Acknowledge.
- MEMR : Memory Read.
- MEMW : Memory write.
- IOR : I/O Read.
- IOW : I/O Write.
- READY : Synchronizes slower peripheral devices with system clock.
- CS : Chip select.
- CLK : Clock
- RESET : Resets the DMA Controller.
- MARK : Outputs a pulse to indicate the Completion of a block transfer.
- DACK : DMA Acknowledge.