Microprocessor & Microcontroller

By Chetna

Microcomputer

- A small, complete computing system built around a microprocessor.
- Contains CPU, memory, I/O ports, storage, and peripherals.
- Used in PCs, laptops, embedded systems.

Microprocessor

- A CPU on a single chip.
- Requires external memory and I/O devices to function.
- Used in computers, industrial control, automation.

Microcontroller

- A single-chip computer.
- Contains CPU, RAM, ROM, I/O ports, timers, ADC, etc.
- Used in embedded applications like appliances, robots, automotive systems.

Comparison

- Microcomputer: Complete system with CPU, memory, I/O
- Microprocessor: Only CPU, needs external components
- Microcontroller: All essential components on a single chip

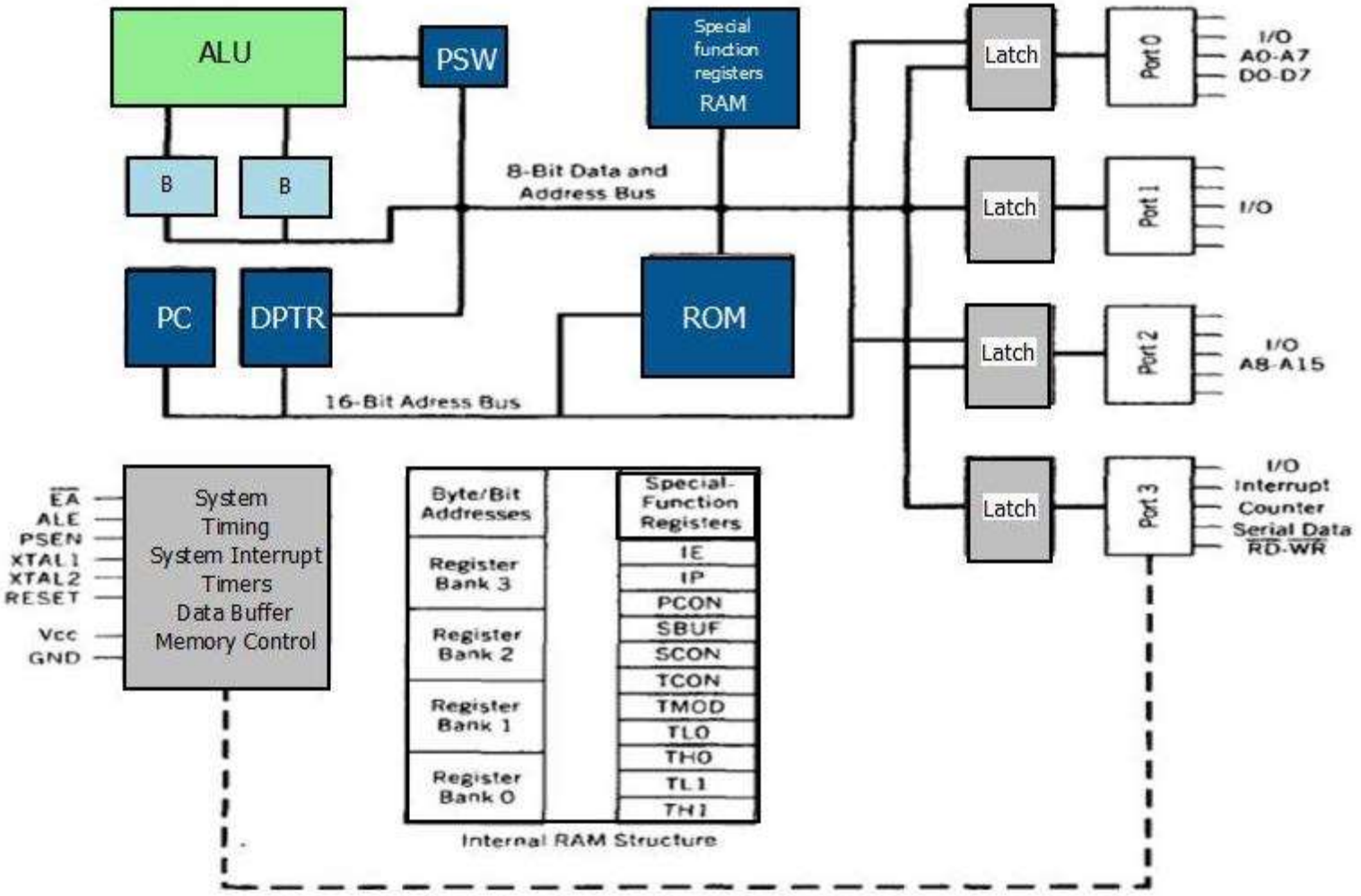
Selection of Microcontroller

- • Application requirements (speed, memory, I/O)
- • Power consumption
- • Cost and availability
- • Peripheral support (ADC, UART, SPI, timers)
- • Development tools and community support

History of 8051

- • Developed by Intel in 1980 as part of the MCS-51 family.
- • 8-bit microcontroller widely used in embedded systems.
- • Contains ROM, RAM, I/O ports, timers, serial port on a single chip.
- • Many enhanced versions developed by Atmel, Philips, Siemens, etc.

8051 Architecture



Architecture Details

- • 8-bit ALU and Accumulator
- • 4 Register Banks
- • 128 bytes RAM + SFRs
- • 4 I/O Ports (P0–P3)
- • Program Counter & Data Pointer
- • Timers, Serial Port, Interrupts

8051 Pin Diagram

Pin diagram of 8051 Microcontroller

Pin Functions

- • Port 0 (P0.0–P0.7): AD0–AD7 Multiplexed Address/Data
- • Port 1 (P1.0–P1.7): General purpose I/O
- • Port 2 (P2.0–P2.7): A8–A15 Address lines
- • Port 3: RXD, TXD, INT0, INT1, T0, T1, RD, WR
- • EA, ALE, PSEN: External memory control
- • Vcc & GND

Crystal Oscillator Circuit

- • XTAL1 and XTAL2 pins connect to quartz crystal.
- • Generates clock frequency for internal timing.
- • Typical frequency: 11.0592 MHz.
- • Two capacitors (20–30 pF) improve stability.

Reset Circuit

- • RESET pin must be high for at least 2 machine cycles.
- • Typically implemented with a capacitor and resistor (RC network).
- • Ensures clean startup when power is applied.

Programming Languages for 8051

- 1. Assembly Language
- • Low-level, hardware-specific
- • Faster execution
- • All instructions written in mnemonics (MOV, ADD, SJMP etc.)
- 2. Embedded C
- • High-level language
- • Supports modular programming
- • Widely used in modern embedded systems
- 3. BASIC / Forth (Rare Usage)
- • Used earlier for simple educational microcontrollers

Advantages of Programming in C

- • Simple, readable, and structured
- • Faster program development
- • Easier debugging & maintenance
- • Code portability across 8051 family
- • Supports functions, arrays, loops, pointers
- • Allows direct access to registers and ports

Addressing Modes of 8051

- 1. Immediate Addressing
- Example: `MOV A, #25H`
- 2. Register Addressing
- Example: `MOV A, R2`
- 3. Direct Addressing
- Example: `MOV A, 30H`
- 4. Register Indirect Addressing
- Example: `MOV A, @R0`
- 5. Indexed Addressing
- Example: `MOVC A, @A+DPTR`

Instruction Set of 8051

- 8051 instructions are divided into:
- • Data transfer
- • Arithmetic
- • Logical
- • Branching
- • Bit manipulation
- • Boolean instructions

Data Transfer Instructions

- Examples:
- • MOV Rd, Rs
- • MOV A, @R0
- • MOVC A, @A+DPTR
- • MOVX A, @DPTR

- Used for transferring data between registers, memory, and I/O.

Arithmetic Instructions

- Examples:
- • ADD A, Rn
- • SUBB A, #data
- • INC A
- • DEC direct
- • MUL AB
- • DIV AB

- Used for mathematical operations.

Logical Instructions

- Examples:
- • ANL A, #data
- • ORL A, Rn
- • XRL A, direct
- • CLR A
- • CPL C

- Used for logic and bit operations.

Branch Instructions

- Examples:
- • SJMP rel
- • LJMP addr
- • CJNE A, #data, rel
- • DJNZ R0, rel

- Used for conditional branching and looping.

Bit Manipulation Instructions

- Examples:
- • SETB bit
- • CLR bit
- • CPL bit
- • JB bit, label
- • JNB bit, label

- Useful for controlling individual pins.

Steps to Generate HEX File in Keil

- 1. Open **Keil uVision**
- 2. Create **New Project**
- 3. Select **8051 Device** (e.g., AT89C51)
- 4. Add C/ASM file
- 5. Write program
- 6. Go to **Project → Build Target**
- 7. HEX file is created in *Objects* folder
- 8. HEX file is used for **burning into microcontroller**

8051 Timers and Registers

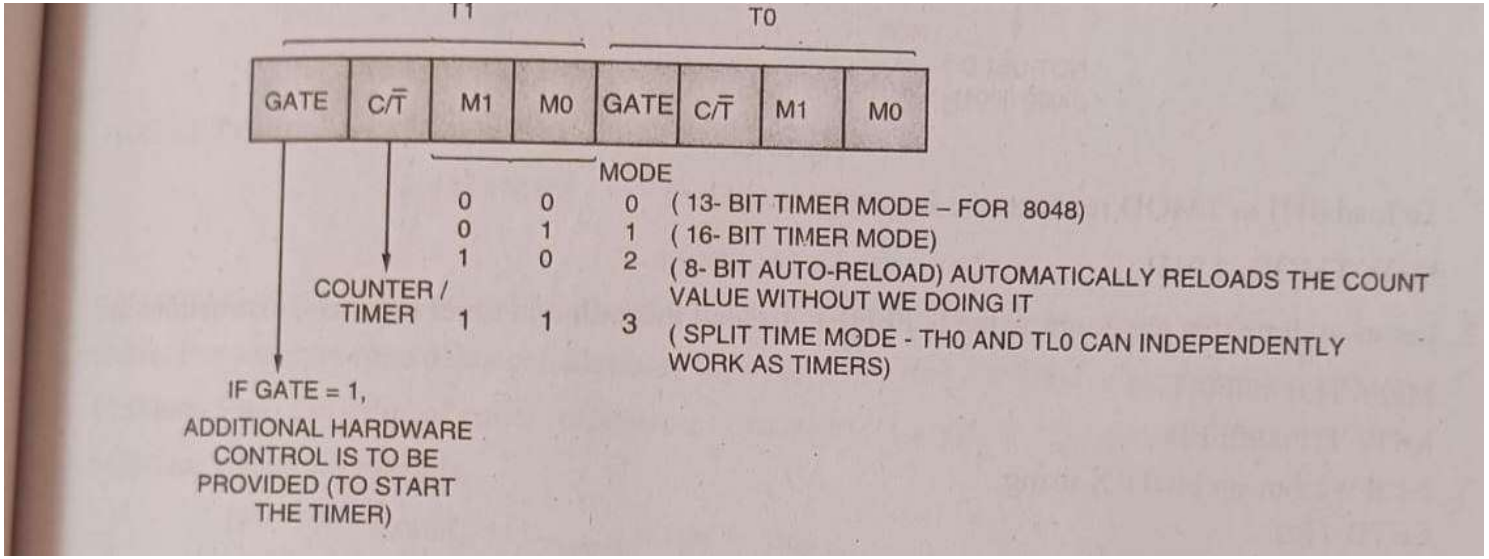
- **8051 has two timers**
- Timer 0
- Timer 1
- Both are 16-bit timers (can count 0–65535).
- **Registers used by timers**
- TH0, TL0 → High & low byte of Timer 0
- TH1, TL1 → High & low byte of Timer 1

8051 Timers and Registers

- **Special Function Registers**
- TMOD (Timer Mode Register)
- Used to select mode & timer/counter operation.
- TCON (Timer Control Register)
- Used to start/stop timers and check overflow flag.

TMOD (Timer Mode Register)

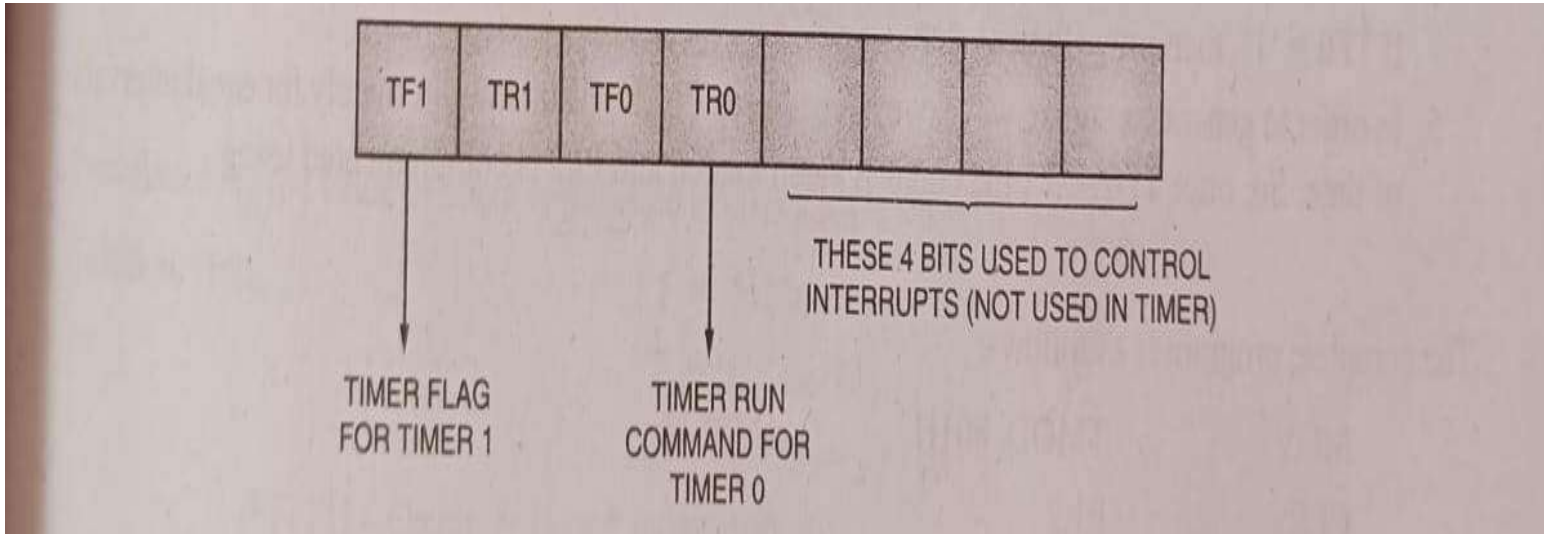
- 1. TMOD. This register is not bit- addressable (individual bits can't be accessed).



- There are two ways to start a timer: Software command or hardware control.

TCON (Timer Control Register)

- 2. TCON. This register is bit-addressable.



Steps for configuration of timer:

- 1. Configure the TMOD register (Select which timer/counter is to be used, in which mode, and whether external hardware control needed or not).
- 2. Load the timer registers (TLX or THX) with the count value.
- 3. Set the bit TRX using instruction SETB TRX. This will help to run the timer.

Steps for configuration of timer:

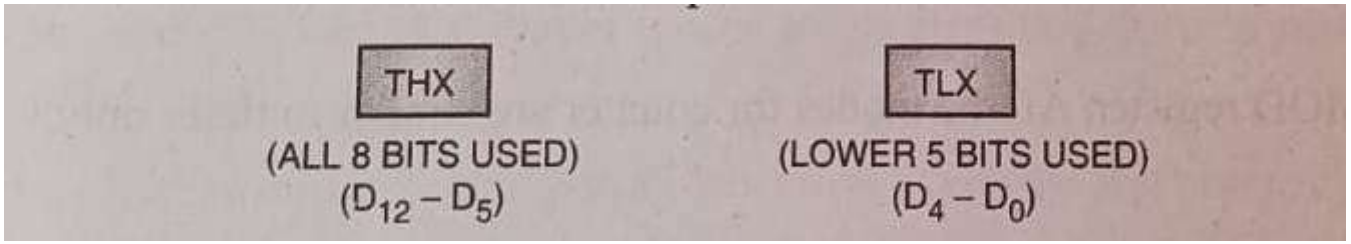
- 4. Let us assume mode 1 is selected. So, the count goes till FFFFH and rolls over to 0000H. Then, the timer flag is automatically set to '1'. The roll over causes an interrupt by making TFX = 1.
- 5. On reaching maximum value, one more pulse sets TFX. So, next step is to check the status of bit TFX.
- 6. If mode 2 (auto reload) is not used, again load the timer registers (Step 2) to repeat the delay process.

Timer/Counter

- **Timer Mode**
- Timer increments using the internal clock (machine cycle frequency).
- **Counter Mode**
- Counter increments when a pulse arrives at external pins:
- T0 → Pin 3.4
- T1 → Pin 3.5

Timer Modes

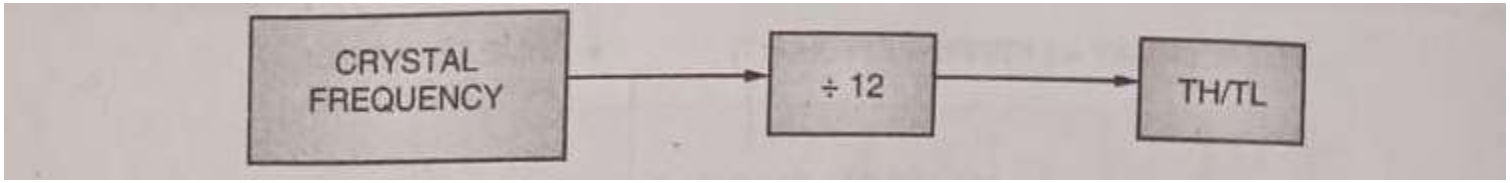
- **Timer Modes**
- 1. Mode 0 – It is the 13-bit timer mode used to make 8051 compatible with 8048. This mode is not used much.



- The range is from 0000H - 1FFFH

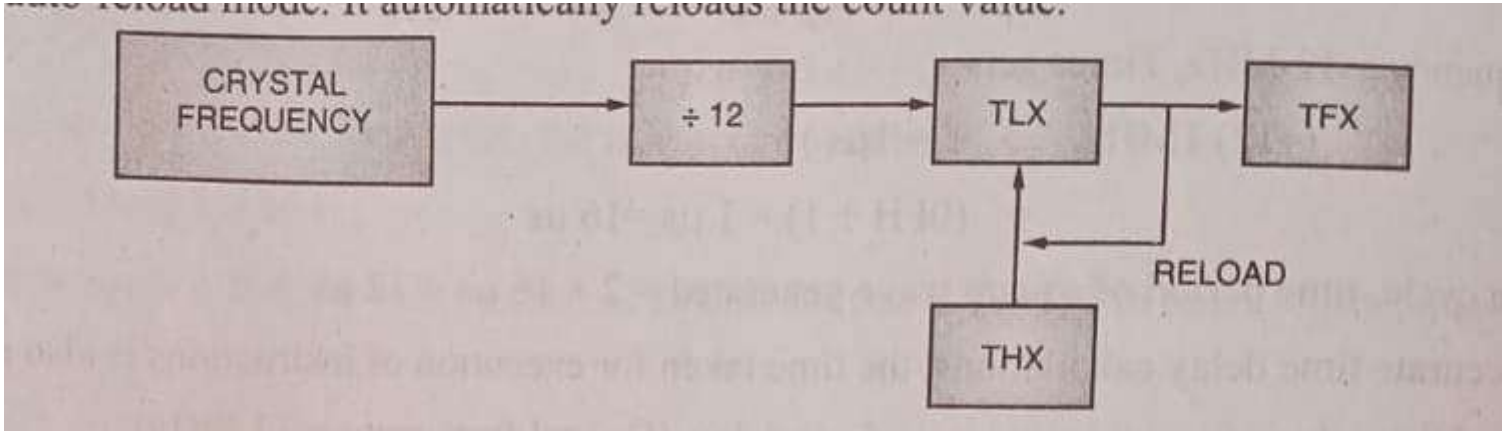
Timer Modes

- 2. Mode 1 – It is the 16-bit timer mode.
- Most commonly used.



Timer Modes

- 3. Mode 2 – It is 8-bit auto-reload mode. It automatically reloads the count value.



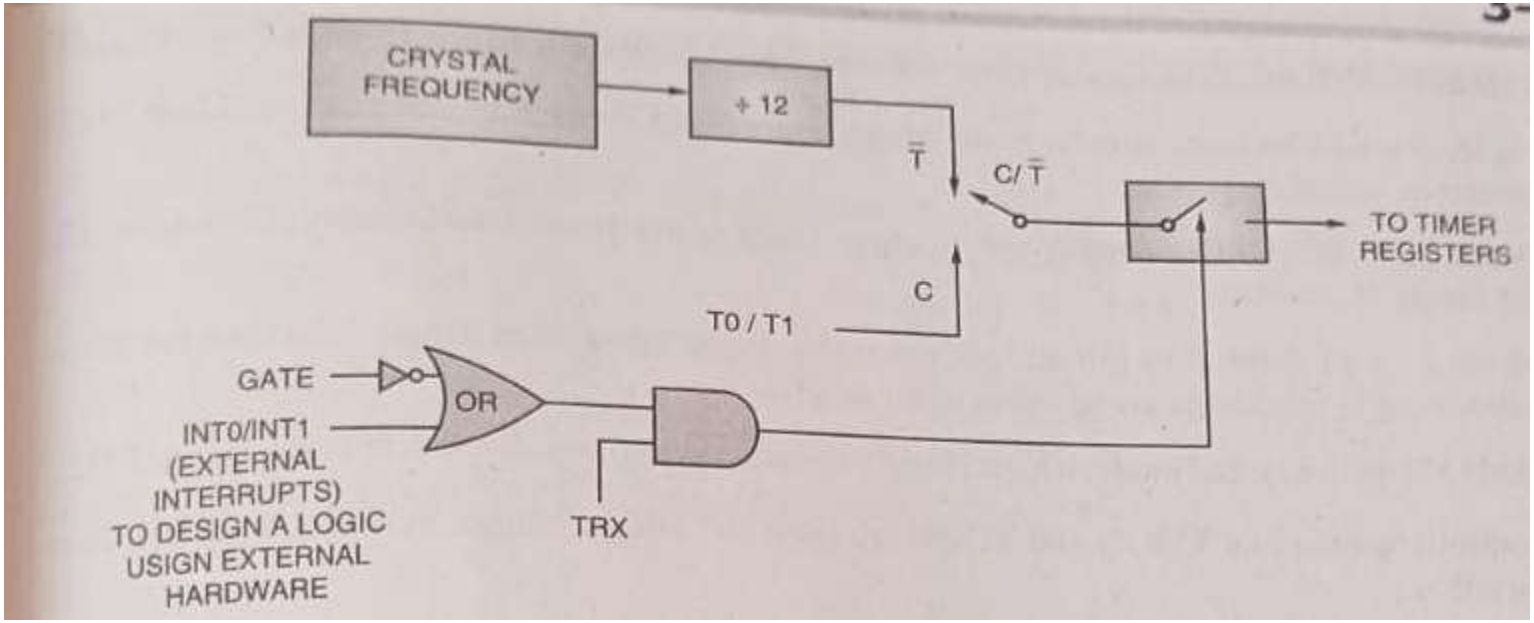
- When TFX is set, after roll over, the reload pulse is generated. THX remains unchanged and TLX is only changed.

Timer Modes

- 4. Mode 3 – It is the split timer mode.
- Here, we have two independent 8-bit timers - TLO (TRO and TFO used) and THO (TR1 and TF1 used), This mode is used in baud rate generation in serial communication.
- In this case there is no TRX or TFX bit for T1. So, T1 can be used as 16 bit timer (8253) only. It will not generate the interrupt. Also, no auto reload option is present.

Counter

- The C/$\bar{T}$ bit 1 in TMOD register. All the modes for counter are similar to timer only.



SERIAL PORT OPERATION

- If the data can be transmitted as well as received simultaneously, it is called full duplex serial communication. In 8051, there is a serial buffer register (8-bits) present, denoted by SBUF. The data to be transmitted is loaded is SBUF. Also, the received data first goes to SBUF.
- There are two special function registers used for programming and configuration in case of serial communication. They are: SCON and PCON.

Baud rate

- It is the number of changes in signal per second. It has to be programmed by the programmer.
- Some of the available options for baud rate are:

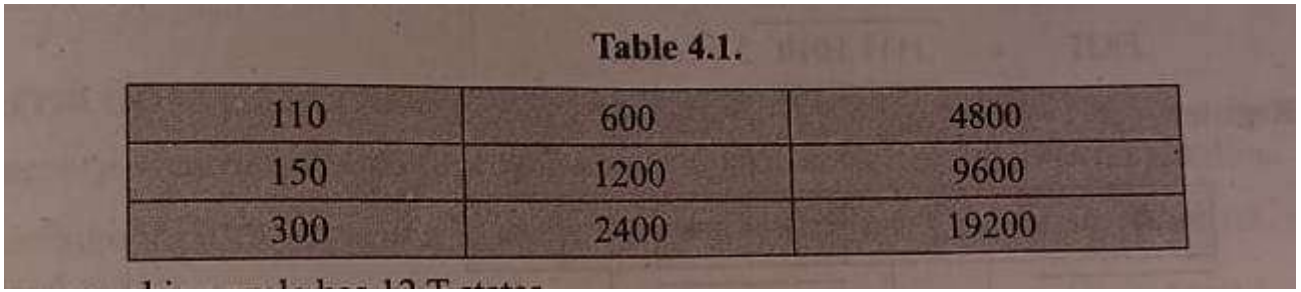
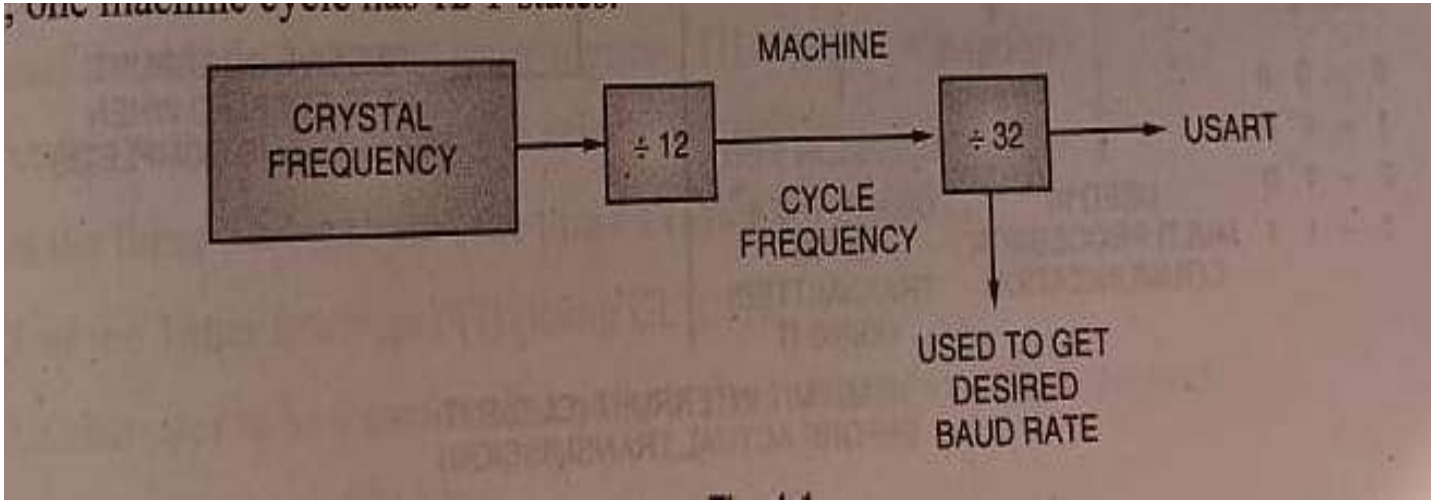


Table 4.1.

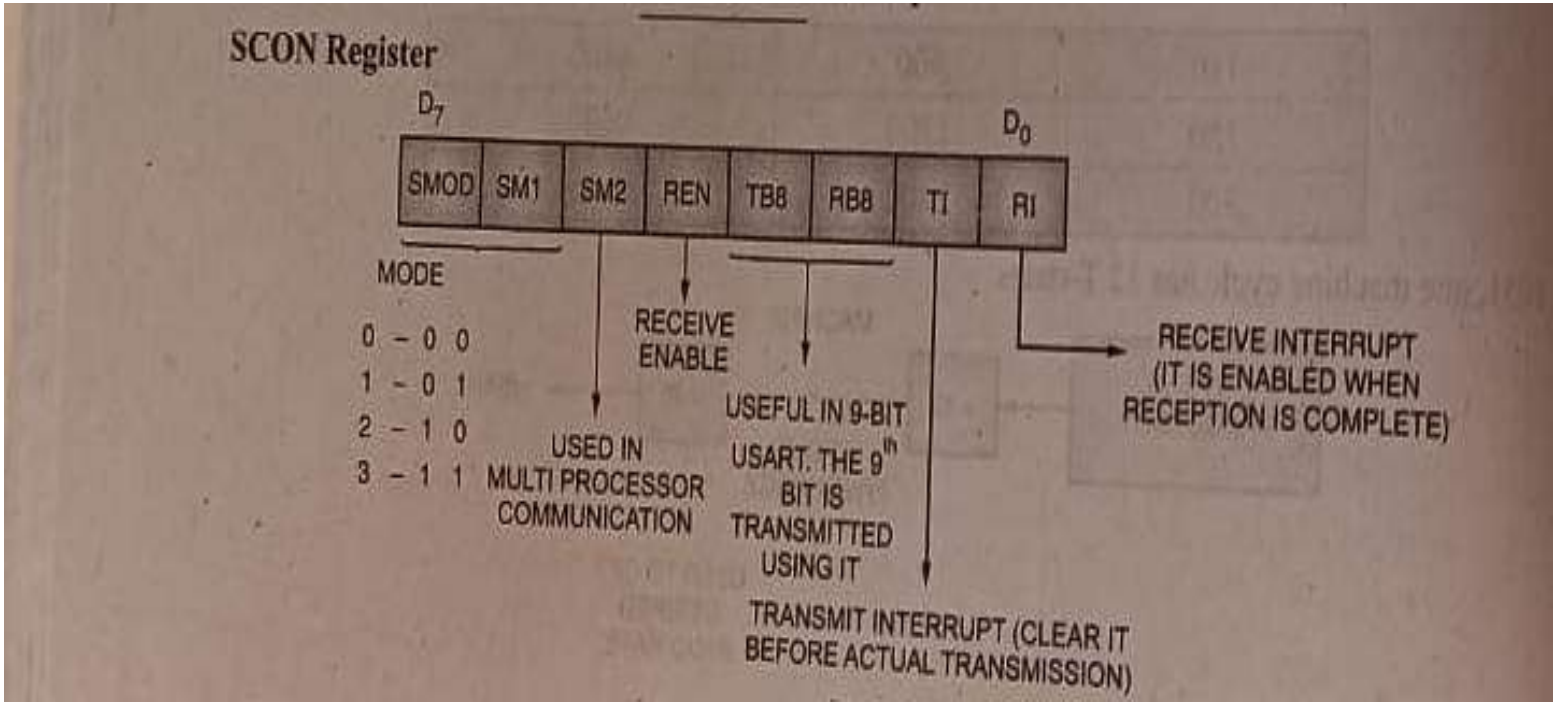
110	600	4800
150	1200	9600
300	2400	19200

Baud rate

- In 8051, one machine cycle has 12 T-states.



SCON Register



SCON Register

- Note. In Asynchronous transmission, TB8 and RB8 bits are used. Data is transmitted in the form of frames.

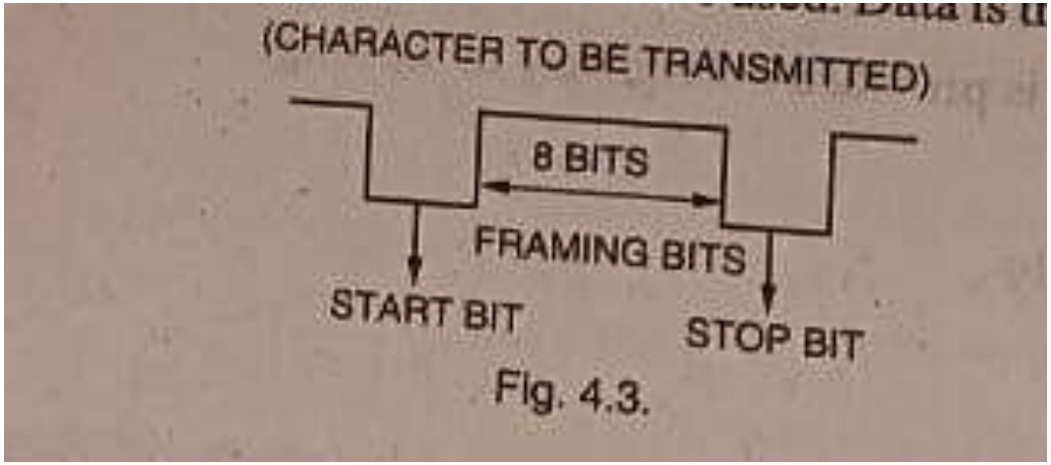


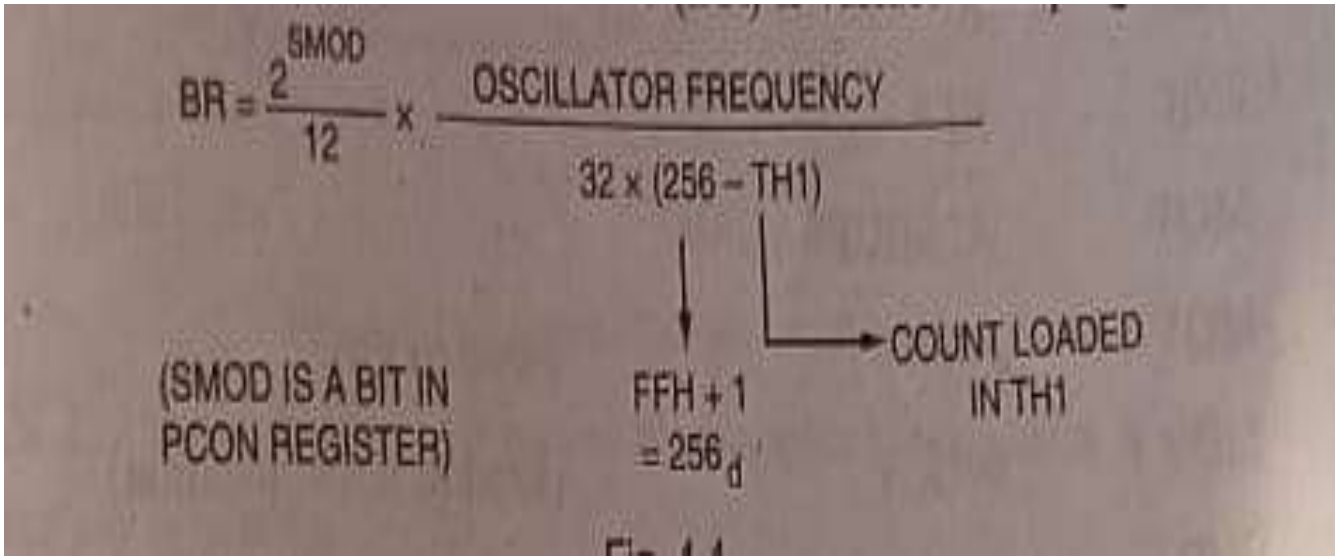
Fig. 4.3.

MODES OF OPERATION

- MODE 0. This mode is for 8-bit USART (User Synchronous Asynchronous Receiver and Transmitter). The Baud Rate is fixed $(f/12)_d$.
- The IC 8251 may be used to convert serial to parallel data or vice versa, if required.

MODES OF OPERATION

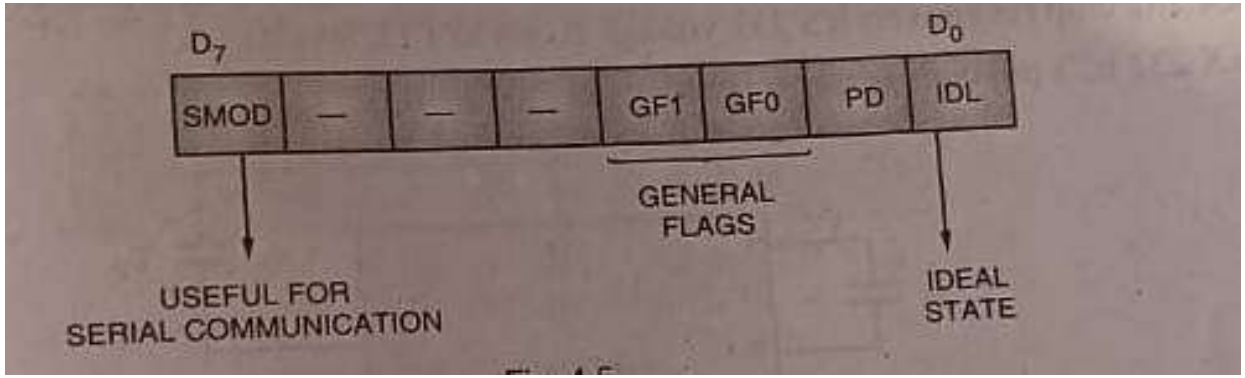
- MODE 1. This mode is 8-bit USART. The baud rate (BR) is variable and programmable.



MODES OF OPERATION

- MODE 2. This mode is 9-bit USART. The BR is fixed here $(f/64)_d$.
- MODE 3. It is 9-bit USART with a variable BR.

PCON Register:



8051 CONNECTION TO RS-232

- RS-232 standard is not TTL compatible. So. 8051 is interfaced to RS-232 connector using a line drive like MAX 232 chip (to convert RS 232 voltage levels to TTL levels).

MAX 232 IC's pin configuration

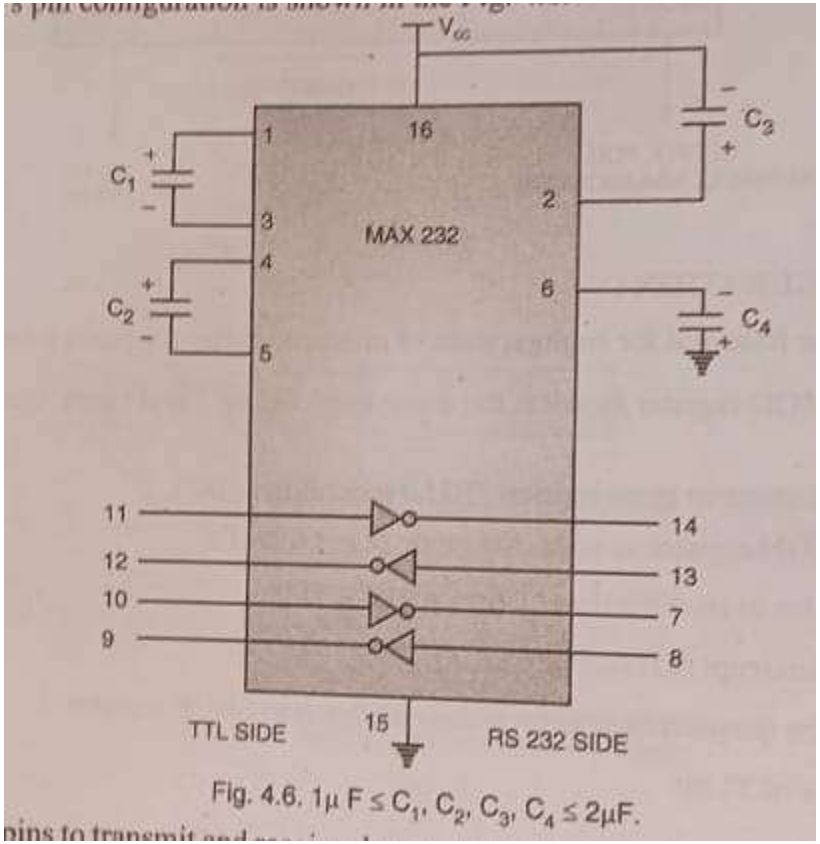


Fig. 4.6. $1\mu F \leq C_1, C_2, C_3, C_4 \leq 2\mu F$.

MAX 232 IC's pin configuration

- **Why needed?**
- Microcontroller/TTL logic uses 0–5V
- RS-232 uses ±12V
- Direct connection would damage the microcontroller.
- **MAX232 Functions:**
- Converts TTL → RS-232 (Tx)
- Converts RS-232 → TTL (Rx)
- Uses charge pump capacitors (C1–C4) to generate ±10V from +5V supply.

RS-232 DB-25 Connector

- RS 232 cable is also called DB-25 connector. There are two types of connectors-DB-25P (plug connector /male) and DB-25S (Socket connector/ female). In PC cables, all 25 pins are not used.

RS-232 DB-25 Connector

- 25-pin connector
- Original standard
- Many pins unused in simple communication
- Common signals:
  - TxD — Transmit Data
  - RxD — Receive Data
  - RTS / CTS — Flow control
  - DTR / DSR — Control lines

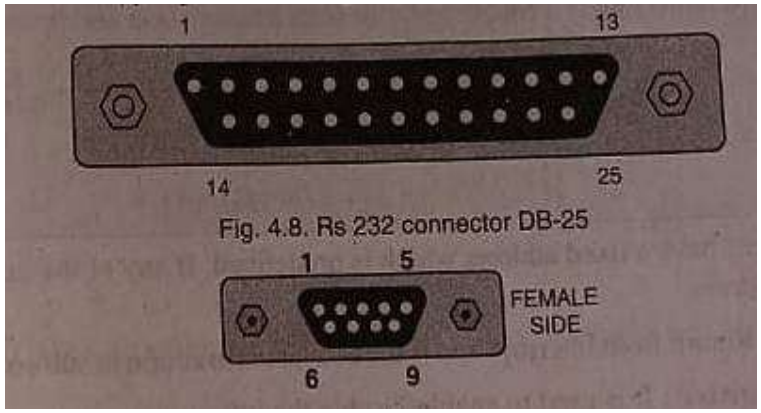


Fig. 4.8. Rs 232 connector DB-25

RS-232 DB-9 Connector

- 9-pin version widely used today
- Essential communication pins:

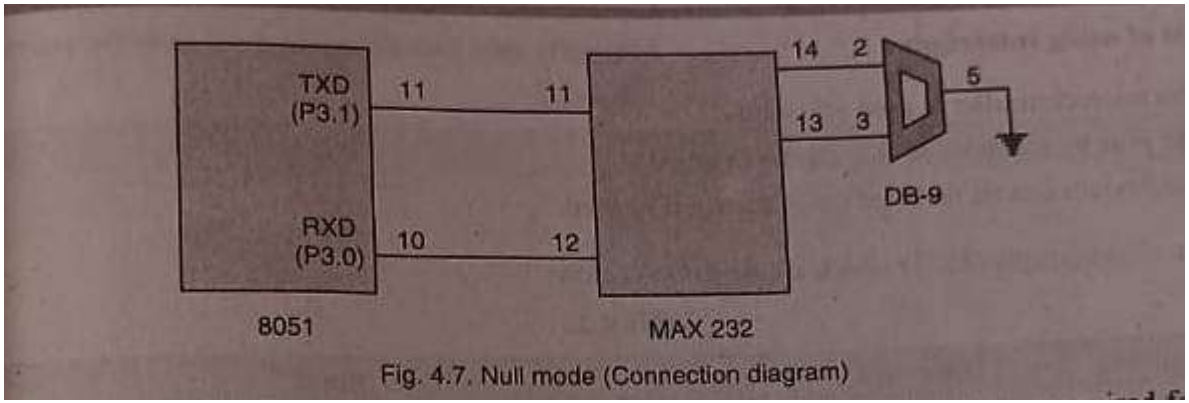
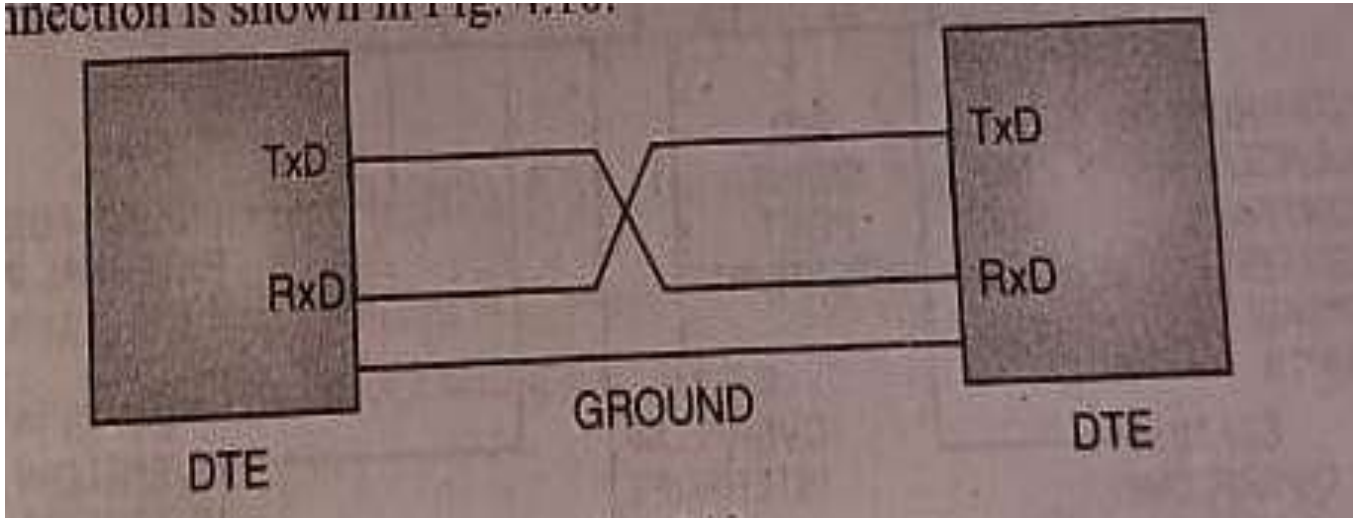


Fig. 4.7. Null mode (Connection diagram)

  - Pin 2: RxD
  - Pin 3: TxD
  - Pin 5: Ground
- Compact and widely supported

Null Modem



Null Modem

- **DTE:** Data Terminal Equipment (Computer, Microcontroller)
- **DCE:** Data Communication Equipment (Modem)
- DTE → DCE direct cable
- DTE → DTE requires null modem
- **Connections:**
- TxD ↔ RxD (crossed)
- RxD ↔ TxD
- Ground ↔ Ground
  Allows direct communication between two computers.

Advantage of MAX 232 chip

- It uses +5V power supply which is the same as one required for 8051. Hence, one power supply is needed.

- Note. MAX 233 chip eliminates the need to use capacitors but its much more expensive than MAX 232 chip.

What Are Interrupts?

- Interrupts are special signals that temporarily stop the main program.
- The microcontroller pauses its current task.
- It executes a small program called **ISR (Interrupt Service Routine)**.
- After finishing ISR, it returns to the main program.

Benefits of Using Interrupts

- **Efficient use of Microcontroller**
- MCU reacts only when needed, no need for continuous monitoring.
- **Priority control**
- Each interrupt can be given priority (high or low).
- **Device masking**
- Any interrupt can be enabled or disabled as required.

Types of Interrupts in 8051

ISR Address	Interrupt	Description
0000H	RESET	It's a signal
0003H	INT0	Pin 3.2 (External Interrupt 0)
000BH	TF0	Timer flag interrupt (Timer 0)
0013H	INT1	External Interrupt (Pin 3.3)
001BH	TF1	Timer 1 (TF1)
0023H	Serial interrupt	Single only for both transmission and reception (RI/TI)

RESET Interrupt (0000H)

- Address: **0000H**
- Occurs when microcontroller is powered ON or reset button pressed.
- CPU starts executing code from this address.
- No ISR—used to begin main program.

INT0 Interrupt (0003H)

- Address: **0003H**
- Triggered by **external signal** at **Pin 3.2**.
- Used for sensing switches, sensors, emergency stop systems.
- Programmer writes ISR starting from this address.

Timer0 Interrupt (000BH)

- Address: **000BH**
- Occurs when **Timer 0 overflows**.
- Used for:
- Generating delays
- Pulse counting
- Repetitive timed tasks

INT1 Interrupt (0013H)

- Address: **0013H**
- Triggered by **external signal** at **Pin 3.3**.
- Works same as INT0 but on separate pin.
- Useful for multi-sensor systems.

Timer1 Interrupt (001BH)

- Address: **001BH**
- Occurs when **Timer 1 overflows**.
- Used for:
- Baud rate generation
- Time-based control
- Repetitive timed functions

Serial Interrupt (0023H)

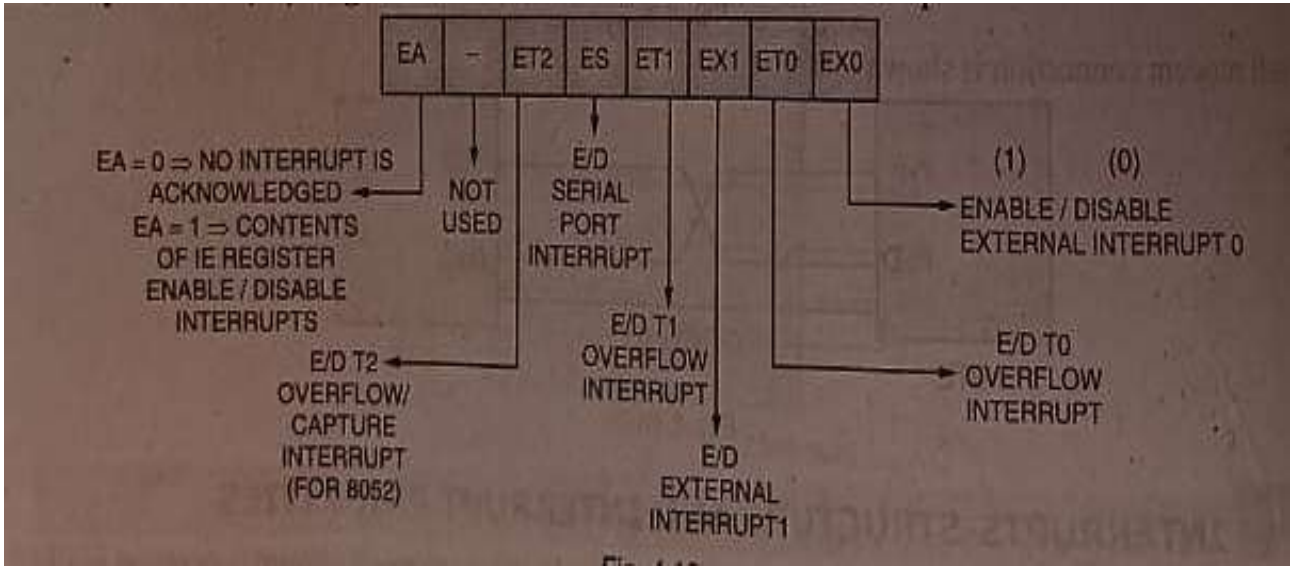
- Address: **0023H**
- Generated when:
- **RI = 1** → Receive interrupt
- **TI = 1** → Transmit interrupt
- Both share the same vector address.
- Used in UART communication.

RETI Instruction

- RETI = **Return from Interrupt**
- Used at the **end of every ISR**.
- Tells microcontroller:
- ISR completed
- Resume main program at the point where it was paused.

Interrupt Enable Register (IE Register)

- IE register controls which interrupts are ON/OFF.

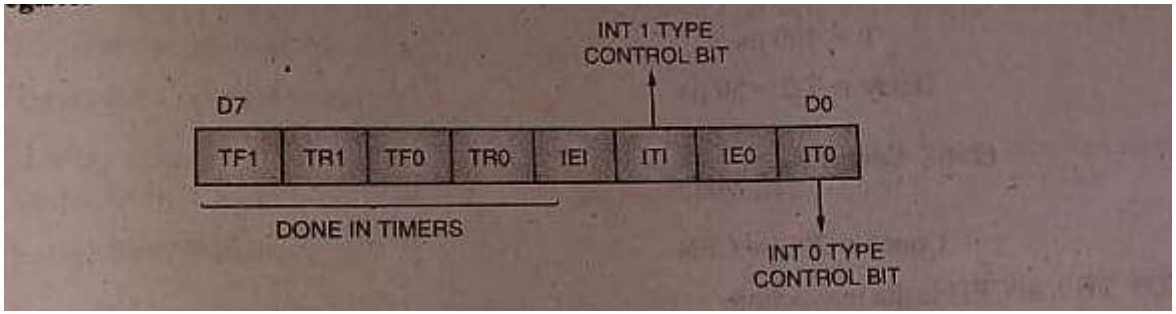


External Interrupts INT0 & INT1

- INT0 → **Pin 12 (Port 3.2)**
- INT1 → **Pin 13 (Port 3.3)**
- Can be triggered by:
- Low-level signal
- Falling edge signal
- Used in emergency stop, sensors, keypress, etc.

TCON Register

- **TCON = Timer Control Register**
- Contains flags for:
- Timer operations (TF0, TF1)
- External interrupts (IE0, IE1)
- Type control bits (IT0, IT1)



IE0 and IE1 – Interrupt Flags

- These are **Interrupt In-Service Flags**.
- Normally **0**.
- Set automatically when an interrupt occurs.
- Used when:
  - Edge-triggered interrupts
  - Level-triggered interrupts (indicates low–high change)
- **Clearing:**
- Cleared automatically when ISR is finished.

IP Register (Interrupt Priority Register)

- IP Register assigns **priority levels** to interrupts.
- Bits in the register:
- How Priority Works?
- Priority determines **which interrupt is served first**.
- Two priority levels:
  - **High**
  - **Low**
- If two interrupts occur simultaneously:
  - The one with **higher priority executes first**.
- Programmers can modify priority through IP register.

What is an LCD?

- LCD = **Liquid Crystal Display**
- Used to display:
  - Numbers
  - Characters
  - Symbols
  - Simple graphics
- Most common: **16×2 LCD (14 pins)**
- Has a built-in controller to refresh the display
- Easy to interface with 8051

LCD Pin Configuration

Pin No.	Name	Description
1	Vss	Ground
2	Vcc	+5V Power supply
3	VEE	Power supply Contrast control
4	RS	Input pin –selects register
5	R/W	Read/Write (0 = Write, 1 = Read)
6	E	Enable I/O pin(high to low pulse, > 450 ns for data
7-14	DB0-DB7	I/O pins 8 bit data bus

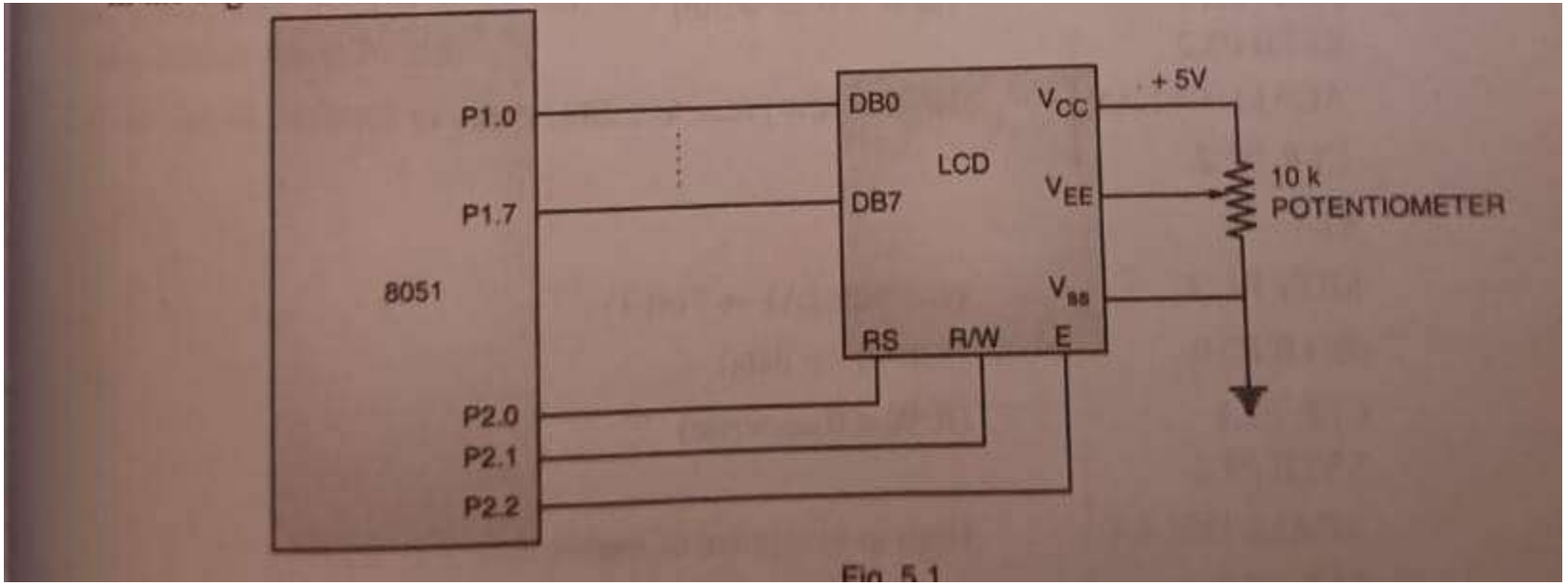
Control Pins

- **1. RS (Register Select)**
- RS = 0 → Command Register selected
- RS = 1 → Data Register selected
- **2. RW (Read/Write)**
- RW = 0 → Write to LCD
- RW = 1 → Read from LCD
- **3. Enable (E)**
- High → Low signal to latch data
- Minimum pulse width: **450 ns**

Data Pins (DB0–DB7

- 8-bit wide data bus
- Used to send:
- Commands
- Characters to be displayed
- Cursor instructions
- In 4-bit mode, only DB4–DB7 are used (your book uses 8-bit mode)

Circuit Diagram for LCD Interfacing



Connections between 8051 and LCD

- **Control Lines**
- **RS → P2.0**
- **RW → P2.1**
- **E → P2.2**
- **Data Lines**
- **DB0–DB7 → Port 1 (P1.0 to P1.7)**
- **Power & Contrast**
- Vcc → +5V
- Vss → GND
- VEE → Potentiometer (controls brightness/contrast)

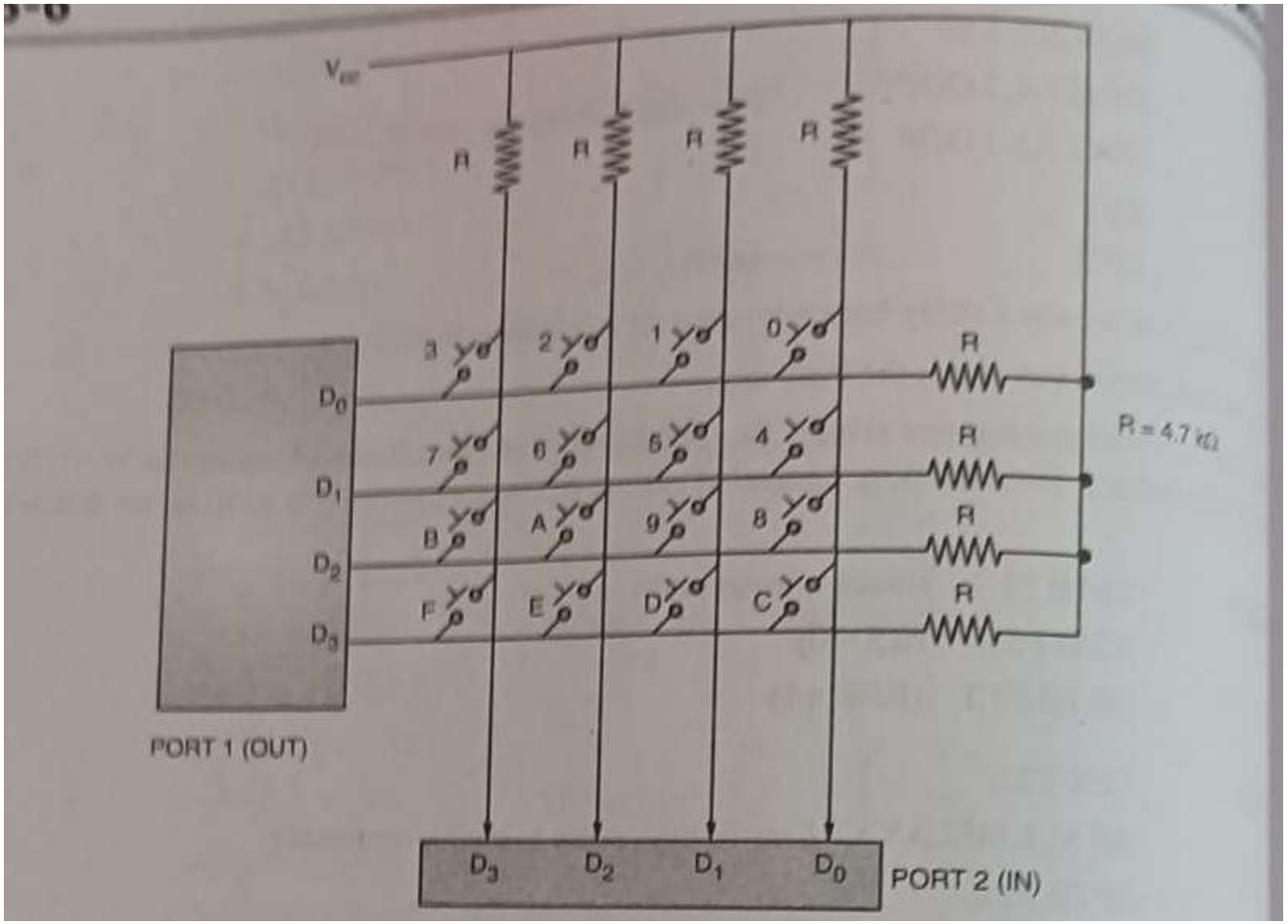
Working of the Circuit

- Data is sent through Port 1
- Commands are selected using RS
- RW decides read/write
- Enable (E) latches the data
- Delay functions ensure the LCD has enough time to process commands
- Potentiometer adjusts display contrast

Two Ways to Program LCD

- **Using Time Delay**
- Simple method
- LCD is assumed ready after delay
- **Checking Busy Flag (BF)**
- Reads DB7 pin
- If BF = 1 → LCD is busy
- More reliable method

Keypad Interfacing



What is a Matrix Keypad?

- A keypad is a set of switches arranged as **rows and columns**.
- A 4×4 keypad has **4 rows × 4 columns = 16 keys**.
- Reduces required MCU pins: only **4 row pins + 4 column pins = 8 pins** needed.
- MCU drives rows (outputs) and reads columns (inputs) — or vice-versa.

Connections

- **Rows → Port 1 (Out)** (example)
- **Columns → Port 2 (In)** (example)
- Pull-ups on column lines (internal or external resistors) so idle read = 1 1 1 1.
- When a key is pressed, a driven row connects to its column and pulls that column low (0).

How it works?

- MCU sets all **rows HIGH** and reads columns → no key pressed ⇒ columns read 1111.
- MCU drives one **row LOW** at a time (others HIGH).
- Read columns:
  - If any column reads 0, a key in the active row is pressed (the column with 0).
- To identify exact key: repeat by testing each row (scan rows sequentially).
- Map (row, column) → specific key label (0–9, A–D, *, # etc).

8051 INTERFACING TO ADC AND DAC

- Understanding how ADC (Analog to Digital Converter) and DAC (Digital to Analog Converter) work with the 8051 microcontroller
- Based on ADC0808/0809 and DAC0808 devices

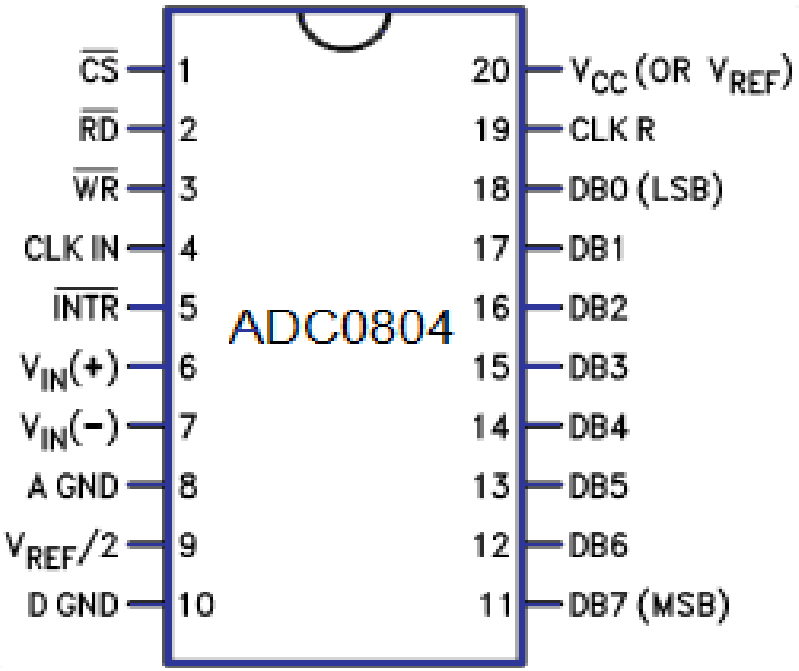
What is an ADC?

- Converts analog voltage into equivalent digital value
- Output is provided to microcontroller via ports
- Examples: ADC0800–ADC0816
- ADC0808 → 8 input channels
- Used in: sensors, measurement systems

Parameters of ADC Conversion

- **1. Conversion Time**
- Depends on clock frequency
- Clock is either:
  ✓ From microcontroller
  ✓ Internal ADC clock
- Formula: f=1/1.1RC
- **2. Step Size**
- Step Size=Vref/ 2^n
- Vref: reference voltage
- n: number of bits (e.g., 8 bits, 10 bits)
- **3. Resolution**
- Smallest change detectable by ADC
- For n bits:
- Resolution = 2^n

Pin Configuration



ADC Pin Diagram Summary

- **CS** → Chip Select
- **RD** → Read Data
- **WR (SOC)** → Start of conversion
- **INTR** → End-of-conversion signal
- **CLKIN** → Clock input
- **AIN(0–7)** → Analog inputs
- **AGND/DGND** → Grounds
- **DB0–DB7** → Digital output (8 bits)

ADC Input Configurations

- Three input modes:
- **Single-ended Mode**
  - One analog input w.r.t ground
- **Pseudo-differential Mode**
  - One signal compared to Vref/2V_{ref}/2Vref/2
- **Differential Mode**
  - Difference between two analog signals

Interfacing ADC with 8051

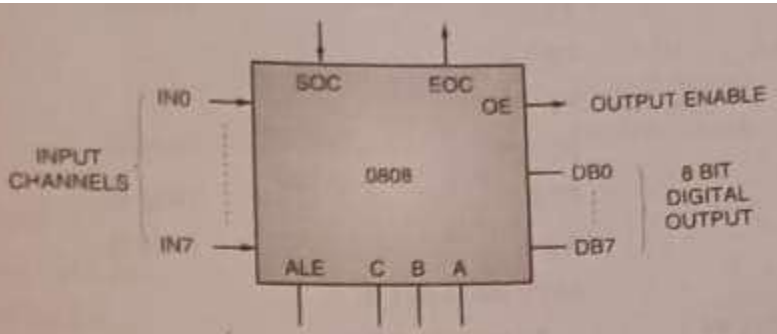


Fig. 5.8

Interfacing of ADC with Microcontroller 8051 (Connection Diagram)

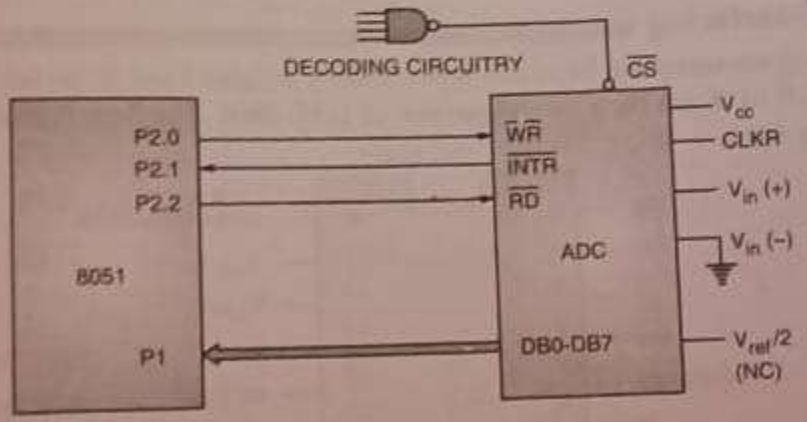


Fig. 5.9.

Interfacing ADC with 8051

- **Connections**
- P2.0 → WR / SOC
- P2.1 → INTR
- P2.2 → RD
- Port P1 ← receives 8-bit ADC output

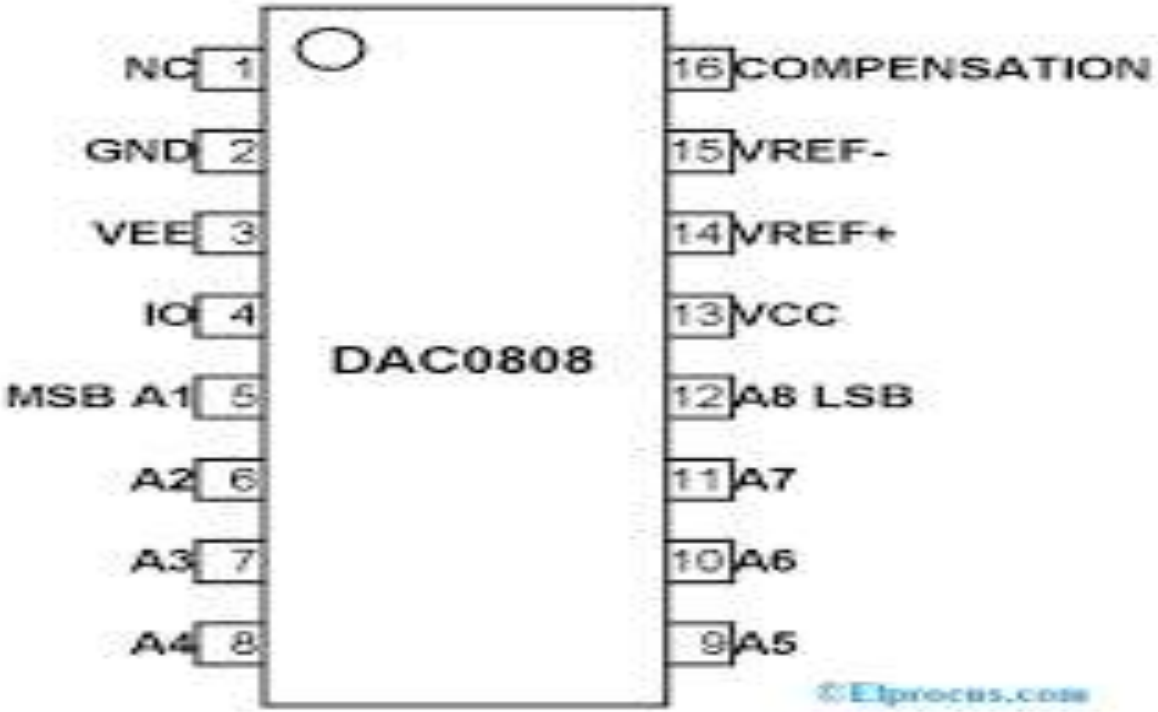
Steps for ADC Interfacing

- Configure **P2.1** as input
- Configure **P1** as input port
- Give SOC pulse via P2.0 (LOW→HIGH)
- Wait until INTR becomes LOW (conversion done)
- Make RD LOW to read data
- Read P1 to get ADC output

What is a DAC?

- **Digital to Analog Converter**
- Converts 8-bit digital data to analog voltage
- Types:
  ✓ Binary Weighted
  ✓ R-2R Ladder (**Used in DAC0808**)

Pin Diagram


NC 1
GND 2
VEE 3
IO 4
MSB A1 5
A2 6
A3 7
A4 8
DAC0808
16 COMPENSATION
15 VREF-
14 VREF+
13 VCC
12 A8 LSB
11 A7
10 A6
9 A5
©Elprocus.com

DAC Pin Details

- Pin1 (NC)- No connection
- Pin2 (GND)-Ground Pin
- Pin3 (VEE)-Negative (-ve) power supply
- Pin4 (IO)-Input/Output signal pin
- Pin5 (A1)- MSB (Digital i/p bit-1)
- Pin6 (A2)-Digital i/p bit-2

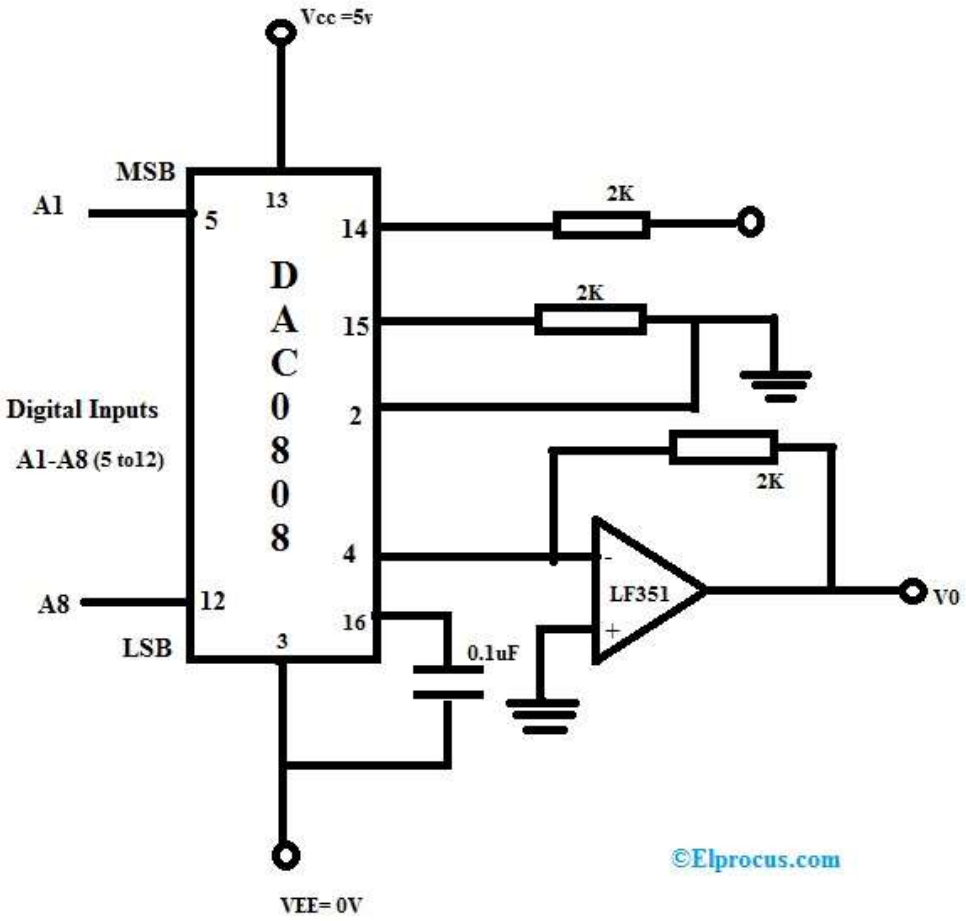
DAC Pin Details

- Pin7 (A3)-Digital i/p bit-3
- Pin8 (A4)-Digital i/p bit-4
- Pin9 (A5)-Digital i/p bit-5
- Pin10 (A6)-Digital i/p bit-6
- Pin11 (A7)-Digital i/p bit-7
- Pin12 (A8)-Digital i/p bit-8 (Least Significant Bit)

DAC Pin Details

- Pin13 (VCC)-Positive (+ve) power supply
- Pin14 (VREF+)-Positive (+ve) reference voltage
- Pin15 (VREF-)-Negative (-ve) reference voltage
- Pin16 (Compensation)-Compensation capacitor pin

Circuit Diagram



Circuit Working

- The IC gets the parallel 8-bit information from **a microcontroller** and changes into an analog signal as an output. The output from the digital to analog or DAC can be an existing quantity as well as this required to be changed into a voltage parameter for applying in several applications very simply.

Circuit Working

- So for changing the present parameter to voltage parameter, an LF351 **operational amplifier** is utilized in the circuit below. Basically, LF351 is one kind of **JFET** op-amp. This is an inexpensive device with the characteristics like high performance, provides high slew rate, & high-gain bandwidth even operating with a low supply.

Circuit Working

- In addition, it includes some features like low current supply, inner compensated i/p off-set voltage. i/p impedance, is high, settling time is fast, harmonic distortion is low. The main application of this IC is while converting digital to analog converters, **S&H circuits**, high-speed integrators, etc.

Circuit Working

- This circuit is known as current to voltage converter. The output from the operational amplifier which is called as an analog voltage is in linear relation among the value of the input digital & therefore the conversion of digital to analog using IC DAC0808 can be attained.