

# **Microprocessors and Microcontrollers Lab**

## **Digital Assessment 5**

**ADARSH A.S.**

**22BKT0058**

**Q1)** Conduct an online survey to explore the 8051 based development boards and their features. (figures and comparison tables may be included)

**Answer:**

exploration of different 8051-based development boards and their features:

**Atmel AT89S52 Development Board:**



Microcontroller: AT89S52

- Clock Speed: 11.0592 MHz
- Memory: 8 KB Flash, 256 Bytes RAM
- I/O Pins: 32
- Communication Interfaces: UART, SPI, I2C
- Programming Language Support: Assembly, C
- Additional Features: On-board LED indicators, reset button, external crystal oscillator support
- Price: Moderate

## Intel MCS-51 Development Board:



- Microcontroller: Various models within MCS-51 family (e.g., 8051, 8031, etc.)
- Clock Speed: Varies by model (typically around 12 MHz)
- Memory: Varies by model (e.g., 4 KB to 64 KB ROM, 128 Bytes to 4 KB RAM) • I/O Pins: Varies by model (e.g., 32 to 64 pins)
- Communication Interfaces: UART, SPI, I2C (varies by board configuration) • Programming Language Support: Assembly, C
- Additional Features: Compatibility with a wide range of MCS-51 microcontrollers, various expansion options available
- Price: Budget-friendly to moderate, depending on features and model

## 8051 Development Board



- Microcontroller: Various 8051-compatible models
- Clock Speed: Typically 12 MHz
- Memory: Varies (e.g., 4 KB to 64 KB Flash, 128 Bytes to 4 KB RAM)
- I/O Pins: Varies (e.g., 32 to 64 pins) • Communication Interfaces: USB, UART, SPI, I2C
- Programming Language Support: Assembly, C
- Additional Features: On-board USB interface for programming and communication, LED indicators, expansion headers

## 8051 Development Board with Wireless Connectivity:



- Microcontroller: Various 8051-compatible models
- Clock Speed: Typically 12 MHz
- Memory: Varies (e.g., 8 KB to 128 KB Flash, 256 Bytes to 8 KB RAM)
- I/O Pins: Varies (e.g., 32 to 64 pins)
- Communication Interfaces: Wireless (Wi-Fi, Bluetooth), UART, SPI, I2C
- Programming Language Support: Assembly, C
- Additional Features: On-board wireless module, antenna, support for IoT applications, external sensor interfaces
- Price: High due to wireless capabilities and additional features

## TABULAR DIFFERENCE

### Comparison of 8051 Development Boards

Feature	Atmel AT89S52 Board	Intel MCS-51 Board	8051 Board (USB)	8051 Board (Wireless)
Microcontroller	AT89S52 (specific)	Various MCS-51 models	Various 8051 models	Various 8051 models
Clock Speed	11.0592 MHz	Varies (typically 12 MHz)	Typically 12 MHz	Typically 12 MHz
Memory (Flash/RAM)	8 KB / 256 Bytes	Varies (e.g., 4 KB-64 KB / 128 Bytes-4 KB)	Varies (e.g., 4 KB-64 KB / 128 Bytes-4 KB)	Varies (e.g., 8 KB-128 KB / 256 Bytes-8 KB)
I/O Pins	32	Varies (e.g., 32-64)	Varies (e.g., 32-64)	Varies (e.g., 32-64)
Communication Interfaces	UART, SPI, I2C	UART, SPI, I2C (varies)	USB, UART, SPI, I2C	Wireless (Wi-Fi, Bluetooth), UART, SPI, I2C
Programming Language Support	Assembly, C	Assembly, C	Assembly, C	Assembly, C
Additional Features	LEDs, reset button, crystal support	Wide model compatibility, expansion options	On-board USB, LEDs, expansion headers	Wireless module, antenna, sensor interfaces
Price	Moderate	Budget-friendly to moderate	Moderate to high	High

**Q2)** Write 8051 ALP to run maximum number of operations simultaneously (timers, ports, serial communication, counter, EXT INT, etc ).

**Procedure:**

- i.) Start up the Keil  $\mu$ Vision Software.
- ii.) Create new  $\mu$ Vision project at required directory.
- iii.) Set the device as 8051 microcontroller (AT89C51).
- iv.) Create new item at Source Group 1 in Target 1.
- v.) Set the file type as ASM file.
- vi.) Continue writing the code for the ALP.
- vii.) Translate and build the file.
- viii.) Start debug session, and run code line by line to get output
- ix.) Check output at the memory location set, in memory 1.

**Algorithm:**

**1) Initialization**

Initialize R4 to 00H. Configure Timer 0 and Timer 1 in mode 2 (8-bit auto-reload mode). Set P0 as an input port with pull-ups enabled. Set the baud rate for serial communication. Enable interrupts for Timer 0 overflow (TF0), Timer 1 overflow (TF1), and serial communication (RI and TI).

## **2) Main Loop (BACK label):**

Read data from Port 0 (P0). Send the data to the serial buffer (SBUF). Copy the data to Port 1 (P1). Repeat the loop indefinitely.

## **3) Interrupt Service Routines (ISRs):**

External Interrupt 0 (ORG 000BH): Toggles P3.7.

Timer 0 Interrupt (ORG 0013H): Increments R0, sends its value to Port 2 (P2), and sets P3.3.

Serial Interrupt (ORG 23H): Handles serial data transmission.

## **CODE:**

```

1  ORG 0000H
2  LJMP MAIN
3
4  ORG 000BH
5  CPL P3.7
6  RETI
7
8  ORG 0013H
9  INC R0
10 MOV A,R0
11 MOV P2,A
12 SETB P3.3
13 RETI
14
15 ORG 23H
16 LJMP SERIAL
17
18 ORG 0030H
19 MAIN:
20 MOV R4,#00H
21 MOV TMOD,#22H
22 MOV P0,0FFH
23 MOV TH0,#-92
24 MOV TH1,#0FDH
25 MOV SCON,#50H
26 MOV IE,#97H
27 SETB TR0
28 SETB TR1
29
30 BACK: MOV A,P0
31 MOV SBUF,A
32 MOV P1,A
33 SJMP BACK
34 SERIAL:
35 JB TI,TRANS
36 MOV A,SBUF
37 CLR RI
38 RETI
39 TRANS:
40 CLR TI
41 RETI
42 END

```

# OUTPUT:

The screenshot displays a microcontroller simulator interface. At the top, the assembly code is shown, with lines 29 through 35 highlighted in green. Below the code, the Logic Analyzer window shows a timing diagram for port P3, with a signal transition at 163.4425µs. The Disassembly window shows the current instruction: `35 JB TI,TRANS`. Four I/O port status windows are visible: Parallel Port 1 (P1: 0x07), Parallel Port 0 (P0: 0x0F), Parallel Port 2 (P2: 0x05), and Parallel Port 3 (P3: 0x7F). The Command window at the bottom shows the execution log, including the command `LA P0` and an error message: `*** error 65: access violation at C:\0x0003 : no 'execute/read' permission`. The status bar at the bottom indicates the simulation is running at 11:16:67.203605 sec.

