**DEPT. OF ELECTRICAL & ELECTRONICS ENGINEERING**

**SRM UNIVERSITY, Kattankulathur – 603203.**

|  |
| --- |
| Title of Experiment :9. Seven segment display |
| Name of the candidate : GAUTAM NAG  Register Number :RA1811005010278  Date of Experiment :05-04-2021  Date of submission **:05-04-2021** |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO:** | **MARKS SPLIT UP** | **MAXIMUM MARKS (50)** | **MARKS OBTAINED** |
| 1 | PRE LAB | 5 |  |
| 2 | PROGRAM | 25 |  |
| 3 | EXECUTION | 15 |  |
| 4 | POST LAB | 5 |  |
| TOTAL | | 50 |  |

**Staff Signature**

**9. Seven segment display**

**PRE-LAB**

1. **There are different modes that can be used for each timer what are they?**

The timer registers can be used in two modes. These modes are Timer mode and the Counter mode. The only difference between these two modes is the source for incrementing the timer registers.

1. **What is the equivalent of the instruction SETB TCON.6?**

SET B TRl

1. **What is the function of the ANL C, bit?**

The ANL instruction performs a bitwise logical AND operation between the specified byte or bit operands and stores the result in the destination operand. When this instruction is used to modify an output port, the value used as the port data will be read from the output data latch, not the input pins of the port.

1. **What is LCALL and ACALL?**

ACALL is a 2-byte instruction in contrast to LCALL, which is 3 bytes. ... The only difference is that the target address for LCALL can be anywhere within the 64K-byte address space of the 8051 while the target address of ACALL must be within a 2K-byte range

1. **What is the function of the instruction MOVC A,@A+DPTR?**
2. MOVC moves a byte from Code Memory into the Accumulator. The Code Memory address from which the byte will be moved is calculated by summing the value of the Accumulator with either DPTR or the Program Counter (PC)
3. It is used by the 8051 to access external memory using the address indicated by DPTR

**9. Seven segment display**

**Aim:**

To write an assembly language program to display characters on a seven segment display

Blinking an LED using EdSim 51.

**Apparatus required:**

8051 microcontroller kit

(0-5V) DC battery

**Algorithm:**

1. Enter a program.

2. Initialize number of digits to Scan

3. Select the digit position through the port address C0

4. Display the characters through the output at address C8.

5. Check whether all the digits are display.

6. Repeat the Process.

**PROGRAM:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Memory Location** | **Label** | **Opcode** | **Mnemonics** | **Comments** |
| 4100 | START | 90 45 00 | MOV DPTR, #address | Data to be displayed |
| 4103 |  | AA 82 | MOV R2, DPL |  |
| 4105 |  | AB 83 | MOV R3, DPH |  |
| 4107 |  | 78 07 | MOV R0, #07H | total digit positions in seven display |
| 4109 |  | 7F 08 | MOV R7, #08H | Initialize no.of digits to scan |
| 410B | L1 | E8 | MOV A, R0 | Select digit position |
| 410C |  | 90 FF C0 | MOV DPTR, #FFC0H |  |
| 410F |  | F0 | MOVX @DPTR, A |  |
| 4110 |  | 8A 82 | MOV DPL, R2 |  |
| 4112 |  | 8B 83 | MOV DPH, R3 |  |
| 4114 |  | E0 | MOVX A, @DPTR |  |
| 4115 |  | 90 FF C8 | MOV DPTR, #FFC8H |  |
| 4118 |  | F0 | MOVX @DPTR, A |  |
| 4119 |  | 12 41 22 | LCALL DELAY |  |
| 411C |  | 0A | INC R2 |  |
| 411D |  | 18 | DEC R0 | Check if 8 digits are displayed |
| 411E |  | DF EB | DJNZ R7, L1 | If not repeat |
| 4120 |  | 21 00 | AJMP START | Repeat from the 1st digit |
| 4122 | DELAY | 7C 02 | MOV R4, #02H |  |
| 4124 | L3 | 7D FF | MOV R5, #FFH |  |
| 4126 | L2 | DD FE | DJNZ R5, L2 |  |
| 4128 |  | DC FA | DJNZ R4, L3 |  |
| 412A |  | 22 | RET |  |

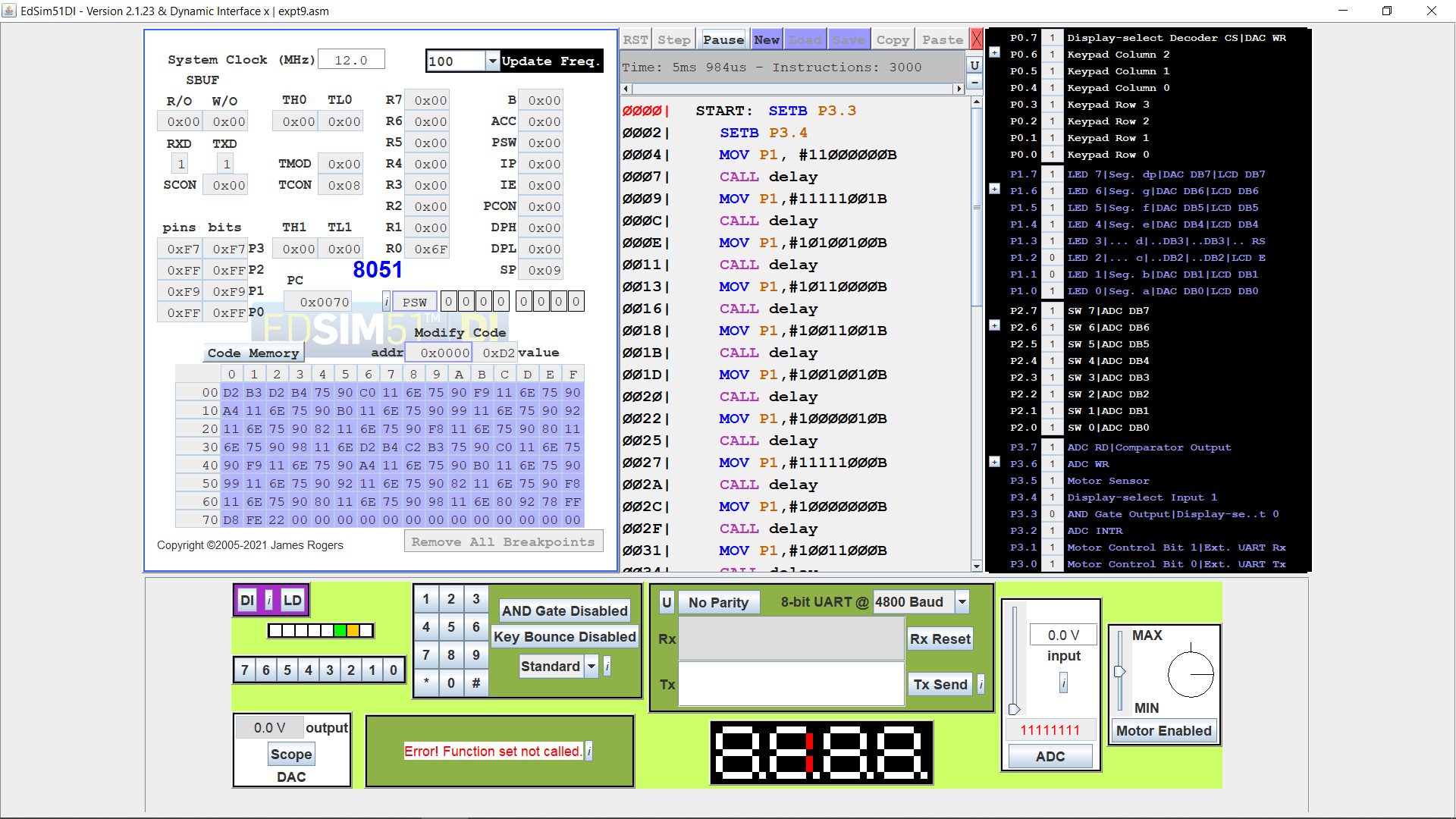
**EDSIM51 PROGRAM-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ADDRESS** | **LABEL** | **MNEMONICS** | **OPCODE** | **COMMENTS** |
| **0000** | **START** | SETB P3.3 | 90 45 00 | **Setup pin 3.3** |
| **0002** | **CLEAR** | SETB P3.4 | AA 82 | **Setup pin 3.4** |
| **0004** |  | MOV P1,#11000000B | AB 83 | **this is set 0 of number ‘1’ of the 7 segment display** |
| **0007** |  | CALL delay | 78 07 | **Specify delay** |
| **0009** |  | MOV P1,#11111001B | 7F 08 | **this is set 0 of number ‘1’ of the 7 segment display** |
| **000C** |  | CALL delay | E8 | **Specify delay** |
| **000E** |  | MOV P1,#10100100B | 90 FF C0 | **this is set 0 of number ‘2’ of the 7 segment display** |
| **00011** |  | CALL delay | F0 | **Specify delay** |
| **00013** |  | MOV P1,#10110000B | 8A 82 | **this is set 0 of number ‘3’ of the 7 segment display** |
| **00016** |  | CALL delay | 8B 83 | **Specify delay** |
| **00018** |  | MOV P1,#10011001B | E0 | **this is set 0 of number ‘4’ of the 7 segment display** |
| **0001B** |  | CALL delay | 90 FF C8 | **Specify delay** |
| **0001D** |  | MOV P1,#10010010B | F0 | **this is set 0 of number ‘5’ of the 7 segment display** |
| **00020** |  | CALL delay | 12 41 22 | **Specify delay** |
| **00022** |  | MOV P1,#10000010B | 0A | **this is set 0 of number ‘6’ of the 7 segment display** |
| **00025** |  | CALL delay | 18 | **Specify delay** |
| **00027** |  | MOV P1,#11111000B | DF EB | **this is set 0 of number ‘7’ of the 7 segment display** |
| **0002A** |  | CALL delay | 21 00 | **Specify delay** |
| **0002C** |  | MOV P1,#10000000B | 7C 02 | **this is set 0 of number ‘8’ of the 7 segment display** |
| **0002F** |  | CALL delay | 7D FF | **Specify delay** |
| **00031** |  | MOV P1,#10011000B | DD FE | **this is set 0 of number ‘9’ of the 7 segment display** |
| **00034** | **START** | CALL delay | DC FA | **Specify delay between digits pop - ups** |
| **00036** | **CLEAR** | SETB P3.4 | 22 | **Set the pin to 1 for the 2nd digit display** |
| **00028** |  | CLR P3.3 | 90 45 00 | **Clear pin number 3** |
| **0003A** |  | MOV P1,#11000000B | AA 82 | **this is set 1 of number ‘1’ of the 7 segment display** |
| **0003D** |  | CALL delay | AB 83 | **Specify delay** |
| **0003F** |  | MOV P1,#11111001B | 78 07 | **this is set 1 of number ‘1’ of the 7 segment display** |
| **00042** |  | CALL delay | 7F 08 | **Specify delay** |
| **00044** |  | MOV P1,#10100100B | E8 | **this is set 1 of number ‘2’ of the 7 segment display** |
| **00047** |  | CALL delay | 90 FF C0 | **Specify delay** |
| **00049** |  | MOV P1,#10110000B | F0 | **this is set 1 of number ‘3’ of the 7 segment display** |
| **0004C** |  | CALL delay | 8A 82 | **Specify delay** |
| **0004E** |  | MOV P1,#10011001B | 8B 83 | **this is set 1 of number ‘4’ of the 7 segment display** |
| **00051** |  | CALL delay | E0 | **Specify delay** |
| **00053** |  | MOV P1,#10010010B | 90 FF C8 | **this is set 1 of number ‘5’ of the 7 segment display** |
| **00056** |  | CALL delay | F0 | **Specify delay** |
| **00058** |  | MOV P1,#10000010B | 12 41 22 | **this is set 1 of number ‘6’ of the 7 segment display** |
| **0005B** |  | CALL delay | 0A | **Specify delay** |
| **0005D** |  | MOV P1,#11111000B | 18 | **this is set 1 of number ‘7’ of the 7 segment display** |
| **00060** |  | CALL delay | DF EB | **Specify delay** |
| **00062** |  | MOV P1,#10000000B | 21 00 | **this is set 1 of number ‘8’ of the 7 segment display** |
| **00065** |  | CALL delay | 7C 02 | **Specify delay** |
| **00067** |  | MOV P1,#10011000B | 7D FF | **this is set 1 of number ‘9’ of the 7 segment display** |
| **0006A** | **Execute** | CALL delay | DD FE | **Specify delay** |
| **0006C** | **DELAY** | JMP start | DC FA | **Execute the program** |
| **0006E** | **OPERAND** | MOV R0,#0FFH | 22 | **Store value of 00FFH to register R0** |
| **00070** | **RESUME** | HERE DJNZ R0,HERE | 90 45 00 | **Decrement byte indicated by first operand** |
| **00072** |  | RET | AA 82 | **Resume execution from resulting place** |

|  |  |
| --- | --- |
| **IN PUT ADDRESS** | **DATA** |
| **0000 - 0034** | ,#11000000B |
| **003A – 006A** | #10010010B |

|  |  |
| --- | --- |
| **OUTPUT** | **DATA** |
| **Seg1** | **0 1 2 3 4 5 6 7 8 9** |
| **Seg2** | **0 1 2 3 4 5 6 7 8 9** |

**SIMULATION :**



**Result:**

Thus an assembly language program blinking an LED displaying on seven segment display has been executed.

**POST-LAB**

1. **Name some bit addressable register?**

A, B, PSW, IP, IE, ACC, SCON, and TCON

1. **How the baud rate can doubled**

With the fixed crystal frequency, baud rate could be doubled by making SMOD – 1. When the SMOD bit is set to 1, 1/12 of XTAL is divided by 16 (instead of 32) and that is the frequency used by Timer 1 to set the baud rate.

1. **What is TI and RI interrupts?**

8051 has a serial communication port and have related serial interrupt flags (TI/RI). When the last bit (stop bit) of a byte is transmitted, the TI serial interrupt flag is set, and when the last bit (stop bit) of the receiving data byte is received, the RI flag gets set.

1. **What are the rotate instructions which involve with carry?**

The ROTATE instructions are primarily used in arithmetic multiply and divide operations and for serial data transfer.

For example: If A is 0000 1000 = 08H 1. By rotating 08H right : A = 0000 0100 = 04H This is equivalent to dividing by 2.

1. **What is the function of SWAP instruction?**

The primary purpose for these swap instructions is to provide an atomic operation for reading from and writing to memory, which has been used to construct mutual-exclusion mechanisms in software for process synchronization.