

BCD TO ASCII CONVERSION USING 8051

Exp No.: 14

Name: S Vishakan

Date: 21-10-2020

Reg. No: 18 5001 196

AIM:

To write an assembly language program to convert a given BCD value to its corresponding ASCII value using an 8051 microcontroller.

PROGRAM – 1: BCD TO ASCII CONVERSION:

ALGORITHM:

1. Begin.
2. Move the value in R1 to A.
3. Get the lower byte at A by performing logical AND over A & 0F.
4. Add 30h to A.
5. Move A to R4.
6. Move the value in R1 to A.
7. Get the higher byte at A by performing logical AND over A & F0.
8. Swap the lower and higher nibble in A.
9. Add 30h to A.
10. Move A to R3.
11. End.

PROGRAM	COMMENTS
MOV R0, #00	R0 has address of 0x00
MOV A, R1	Moving BCD value to A
ANL A, #0FH	Taking lower byte value of A by doing (byte & 0F)
ADD A, #30H	Add 30H to lower byte to convert it to ASCII
MOV R4, A	Move lower ASCII byte to R4 from A
MOV A, R1	Moving BCD value again to A
ANL A, #0F0H	Taking higher byte value of A by doing (byte & F0)
SWAP A	Swap the lower and higher bytes in A
ADD A, #30H	Add 30H to higher byte to convert it to ASCII
MOV R3, A	Move higher ASCII byte to R3 from A
HALT:	
SJMP HALT	Halt the program with a loop.

SAMPLE I/O SNAPSHOT:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 1 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0x00
0x00	0x00	0x00	0x00	R6	0x00	ACC	0x34
RxD	TxD	TMOD	0x00	R5	0x00	PSW	0x01
1	1	0x00	0x00	R4	0x35	IP	0x00
SCON	0x00	TCON	0x00	R3	0x34	IE	0x00
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x45	DPH	0x00
0xFF	0xFF	0x00	0x00	R0	0x00	DPL	0x00
0xFF	0xFF	P2				SP	0x07
0xFF	0xFF	P1					
0xFF	0xFF	P0					

PC 0x000F PSW 0 0 0 0 0 0 0 1

Modify RAM

Data Memory

addr	0x04	0x00	value
0	00	00	00
1	00	45	00
2	00	34	35
3	00	00	00
4	00	00	00
5	00	00	00
6	00	00	00
7	00	00	00
8	00	00	00
9	00	00	00
A	00	00	00
B	00	00	00
C	00	00	00
D	00	00	00
E	00	00	00
F	00	00	00
10	00	00	00
11	00	00	00
12	00	00	00
13	00	00	00
14	00	00	00
15	00	00	00
16	00	00	00
17	00	00	00
18	00	00	00
19	00	00	00
20	00	00	00
21	00	00	00
22	00	00	00
23	00	00	00
24	00	00	00
25	00	00	00
26	00	00	00
27	00	00	00
28	00	00	00
29	00	00	00
30	00	00	00
31	00	00	00
32	00	00	00
33	00	00	00
34	00	00	00
35	00	00	00
36	00	00	00
37	00	00	00
38	00	00	00
39	00	00	00
40	00	00	00
41	00	00	00
42	00	00	00
43	00	00	00
44	00	00	00
45	00	00	00
46	00	00	00
47	00	00	00
48	00	00	00
49	00	00	00
50	00	00	00
51	00	00	00
52	00	00	00
53	00	00	00
54	00	00	00
55	00	00	00
56	00	00	00
57	00	00	00
58	00	00	00
59	00	00	00
60	00	00	00
61	00	00	00
62	00	00	00
63	00	00	00
64	00	00	00
65	00	00	00
66	00	00	00
67	00	00	00
68	00	00	00
69	00	00	00
70	00	00	00
71	00	00	00
72	00	00	00
73	00	00	00
74	00	00	00
75	00	00	00
76	00	00	00
77	00	00	00
78	00	00	00
79	00	00	00

Copyright ©2005-2016 James Rogers Remove All Breakpo...

RST Step Run New Load Save Copy Paste

Time: 36us - Instructions: 23

```

;8051 ALP TO CONVERT A GIVEN B
0000| MOV    R0, #00    ;R0 has ac
0002| MOV    A, R1      ;Moving B
0003| ANL    A, #0FH     ;Taking lo
0005| ADD    A, #30H     ;Add 30H t
0007| MOV    R4, A      ;Move lowe
0008| MOV    A, R1      ;Moving B
0009| ANL    A, #0F0H    ;Taking hi
000B| SWAP   A          ;Swap the
000C| ADD    A, #30H     ;Add 30H t
000E| MOV    R3, A      ;Move high
000F| SJMP   HALT
HALT:

```

P0.7 1 Display-select Decoder CS|DAC WR
P0.6 1 Keypad Column 2
P0.5 1 Keypad Column 1
P0.4 1 Keypad Column 0
P0.3 1 Keypad Row 3
P0.2 1 Keypad Row 2
P0.1 1 Keypad Row 1
P0.0 1 Keypad Row 0
P1.7 1 LED 7|Seg. dp|DAC DB7|LCD DB7
P1.6 1 LED 6|Seg. g|DAC DB6|LCD DB6
P1.5 1 LED 5|Seg. f|DAC DB5|LCD DB5
P1.4 1 LED 4|Seg. e|DAC DB4|LCD DB4
P1.3 1 LED 3|... d|..DB3|..DB3|.. RS
P1.2 1 LED 2|... c|..DB2|..DB2|LCD E
P1.1 1 LED 1|Seg. b|DAC DB1|LCD DB1
P1.0 1 LED 0|Seg. a|DAC DB0|LCD DB0
P2.7 1 SW 7|ADC DB7
P2.6 1 SW 6|ADC DB6
P2.5 1 SW 5|ADC DB5
P2.4 1 SW 4|ADC DB4
P2.3 1 SW 3|ADC DB3
P2.2 1 SW 2|ADC DB2
P2.1 1 SW 1|ADC DB1
P2.0 1 SW 0|ADC DB0
P3.7 1 ADC RD|Comparator Output
P3.6 1 ADC WR
P3.5 1 Motor Sensor
P3.4 1 Display-select Input 1
P3.3 1 AND Gate Output|Display-se..t 0
P3.2 1 ADC INTR
P3.1 1 Motor Control Bit 1|Ext. UART Rx
P3.0 1 Motor Control Bit 0|Ext. UART Tx

DI LD

1 2 3 AND Gate Disabl...
4 5 6 Key Bounce Disabl...
7 8 9 Standard
* 0 #

U Odd Parity 8-bit UART @ 4800 Baud
Rx Rx Reset
Tx Tx Send

0.0 V output
Scope
DAC

BF 0 AC 0x00 IR 0x00 DR 0x00

0.0 V input
11111111
ADC

MAX
MIN
Motor Enabled

8888

RESULT:

An assembly level program was written to convert a given BCD value to its corresponding ASCII value using an 8051 microcontroller and the output was verified.