Lab 4

27 February 2013

Andrew O’Neil-Smith

Objective-

The objective of this lab was to make a simple calculator using the keypad and seven segment display, with indirect memory addressing.

Equipment used-

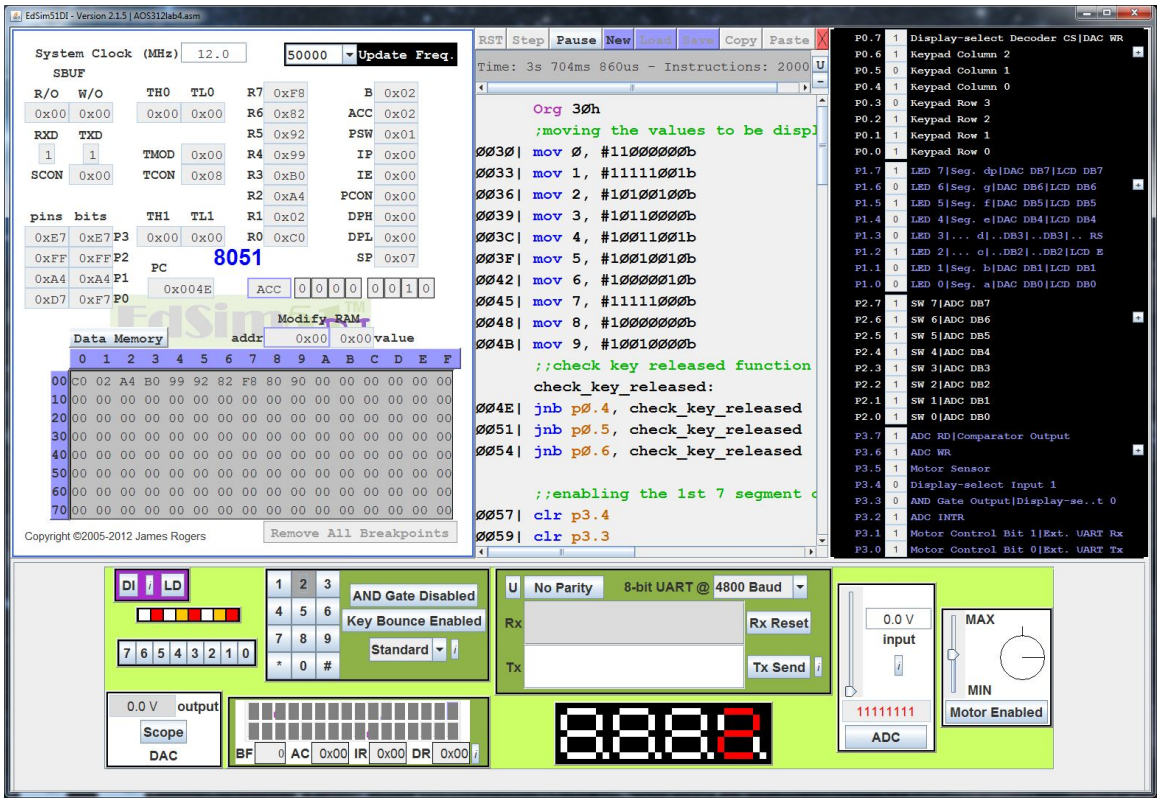
Software: a text editor and an 8051 ASM assembler

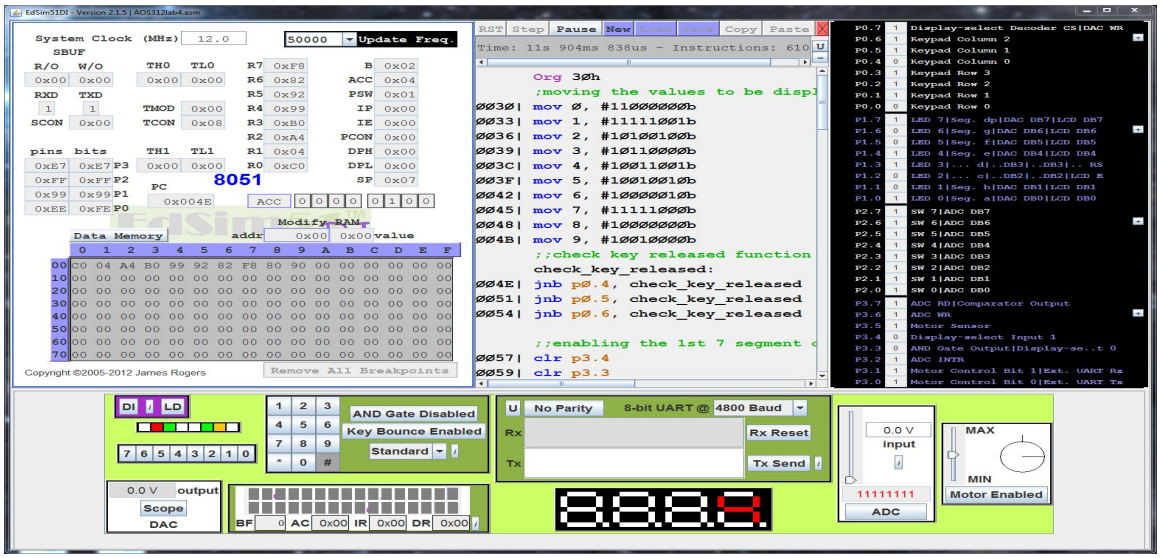
A step debugger that can be used to execute a program one step at a time

Register, code memory, data memory, and input/output port contents are displayed to aid debugging.

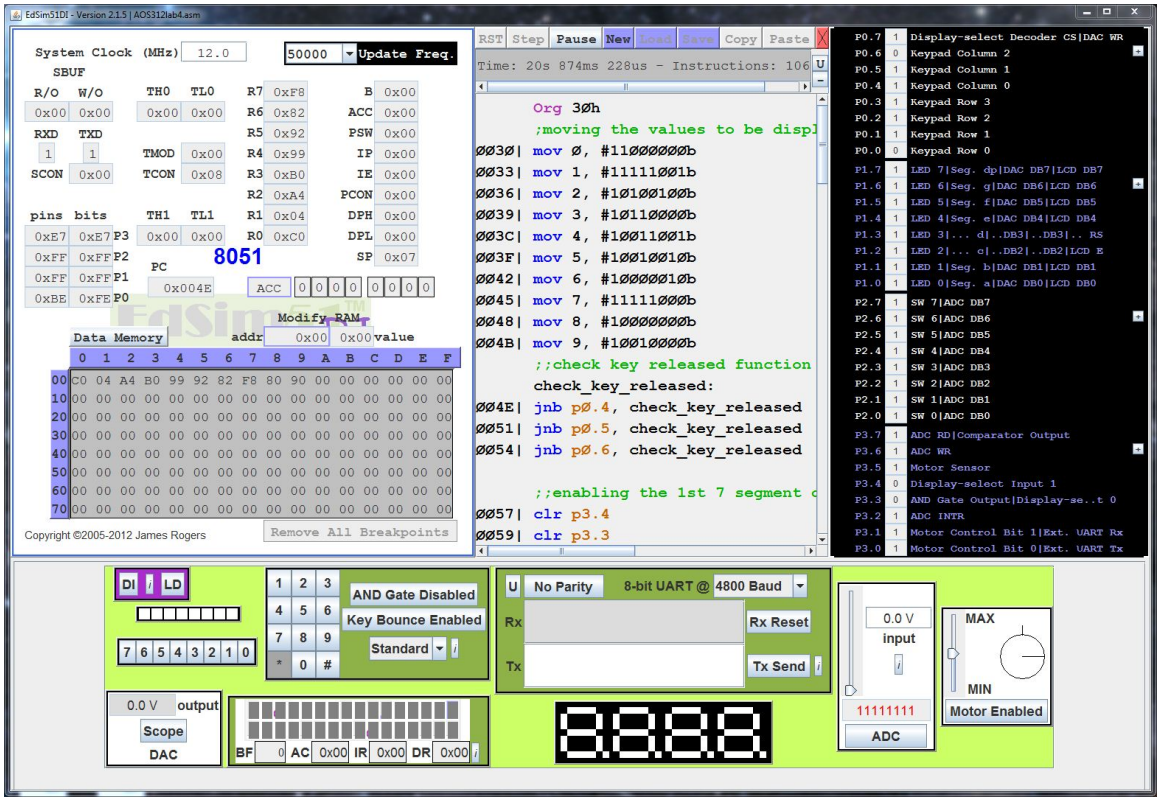
Test Results-

2+2=4




EdSim51DI - Version 2.1.5 | AOS312lab4.asm
System Clock (MHz) 12.0
50000 Update Freq.
SBUF
R/O W/O TH0 TL0 R7 0xF8 B 0x02
0x00 0x00 0x00 0x00 R6 0x82 ACC 0x04
RXD TXD R5 0x92 PSW 0x01
1 1 TMOD 0x00 R4 0x99 IP 0x00
SCON 0x00 TCON 0x08 R3 0xB0 IE 0x00
R2 0xA4 PCON 0x00
pins bits TH1 TL1 R1 0x04 DPH 0x00
0xE7 0xE7 P3 0x00 0x00 R0 0xC0 DPL 0x00
0xFF 0xFF P2 8051 SP 0x07
0xFF 0xFF P1 PC
0xEE 0xFE P0 0x004E ACC 0 0 0 0 0 1 0 0
Modify RAM
Data Memory addr 0x00 0x00 value
0 1 2 3 4 5 6 7 8 9 A B C D E F
00 C0 04 A4 B0 99 92 82 F8 80 90 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Copyright ©2005-2012 James Rogers
Remove All Breakpoints
RST Step Pause New Load Save Copy Paste
Time: 11s 904ms 838us - Instructions: 610
Org 30h
;moving the values to be displ
0030| mov 0, #11000000b
0033| mov 1, #11111001b
0036| mov 2, #10100100b
0039| mov 3, #10110000b
003C| mov 4, #10011001b
003F| mov 5, #10010010b
0042| mov 6, #10000010b
0045| mov 7, #11111000b
0048| mov 8, #10000000b
004B| mov 9, #10010000b
;;check key released function
check_key_released:
004E| jnb P0.4, check_key_released
0051| jnb P0.5, check_key_released
0054| jnb P0.6, check_key_released
;;enabling the 1st 7 segment
0057| clr P3.4
0059| clr P3.3
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 0 Display-select Input 1
P3.3 0 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
7 6 5 4 3 2 1 0
1 2 3
4 5 6
7 8 9
* 0 #
AND Gate Disabled
Key Bounce Enabled
Standard
U No Parity 8-bit UART @ 4800 Baud
Rx
Tx
Rx Reset
Tx Send
0.0 V input
11111111
ADC
MAX
MIN
Motor Enabled
0.0 V output
Scope
DAC
BF 0 AC 0x00 IR 0x00 DR 0x00

Clear


EdSim51DI - Version 2.1.5 | AOS312lab4.asm
System Clock (MHz) 12.0
50000 Update Freq.
SBUF
R/O W/O TH0 TL0 R7 0xF8 B 0x00
0x00 0x00 0x00 0x00 R6 0x82 ACC 0x00
RXD TXD R5 0x92 PSW 0x00
1 1 TMOD 0x00 R4 0x99 IP 0x00
SCON 0x00 TCON 0x08 R3 0xB0 IE 0x00
R2 0xA4 PCON 0x00
pins bits TH1 TL1 R1 0x04 DPH 0x00
0xE7 0xE7 P3 0x00 0x00 R0 0xC0 DPL 0x00
0xFF 0xFF P2 8051 SP 0x07
0xFF 0xFF P1 PC
0xBE 0xFE P0 0x004E ACC 0 0 0 0 0 0 0 0
Modify RAM
Data Memory addr 0x00 0x00 value
0 1 2 3 4 5 6 7 8 9 A B C D E F
00 C0 04 A4 B0 99 92 82 F8 80 90 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Copyright ©2005-2012 James Rogers
Remove All Breakpoints
RST Step Pause New Load Save Copy Paste
Time: 20s 874ms 228us - Instructions: 106
Org 30h
;moving the values to be displ
0030| mov 0, #11000000b
0033| mov 1, #11111001b
0036| mov 2, #10100100b
0039| mov 3, #10110000b
003C| mov 4, #10011001b
003F| mov 5, #10010010b
0042| mov 6, #10000010b
0045| mov 7, #11111000b
0048| mov 8, #10000000b
004B| mov 9, #10010000b
;;check key released function
check_key_released:
004E| jnb P0.4, check_key_released
0051| jnb P0.5, check_key_released
0054| jnb P0.6, check_key_released
;;enabling the 1st 7 segment
0057| clr P3.4
0059| clr P3.3
P0.7 1 Display-select Decoder CS|DAC WR
P0.6 0 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 0 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 0 Display-select Input 1
P3.3 0 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
7 6 5 4 3 2 1 0
1 2 3
4 5 6
7 8 9
* 0 #
AND Gate Disabled
Key Bounce Enabled
Standard
U No Parity 8-bit UART @ 4800 Baud
Rx
Tx
Rx Reset
Tx Send
0.0 V input
11111111
ADC
MAX
MIN
Motor Enabled
0.0 V output
Scope
DAC
BF 0 AC 0x00 IR 0x00 DR 0x00

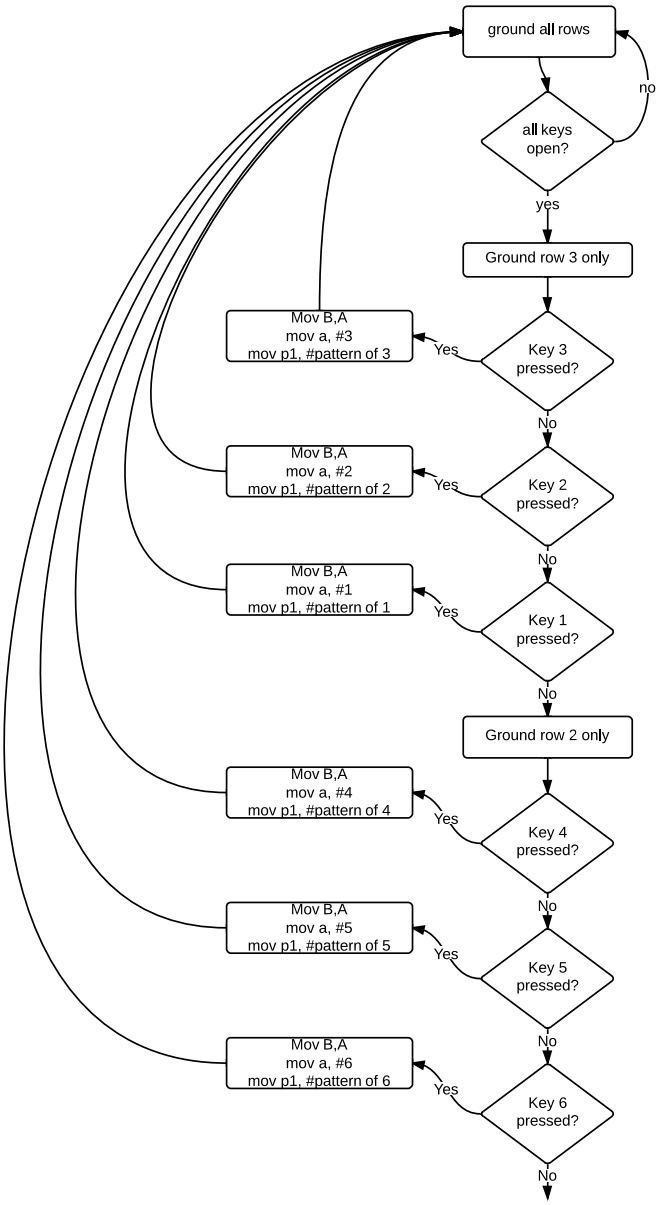
Conclusion-

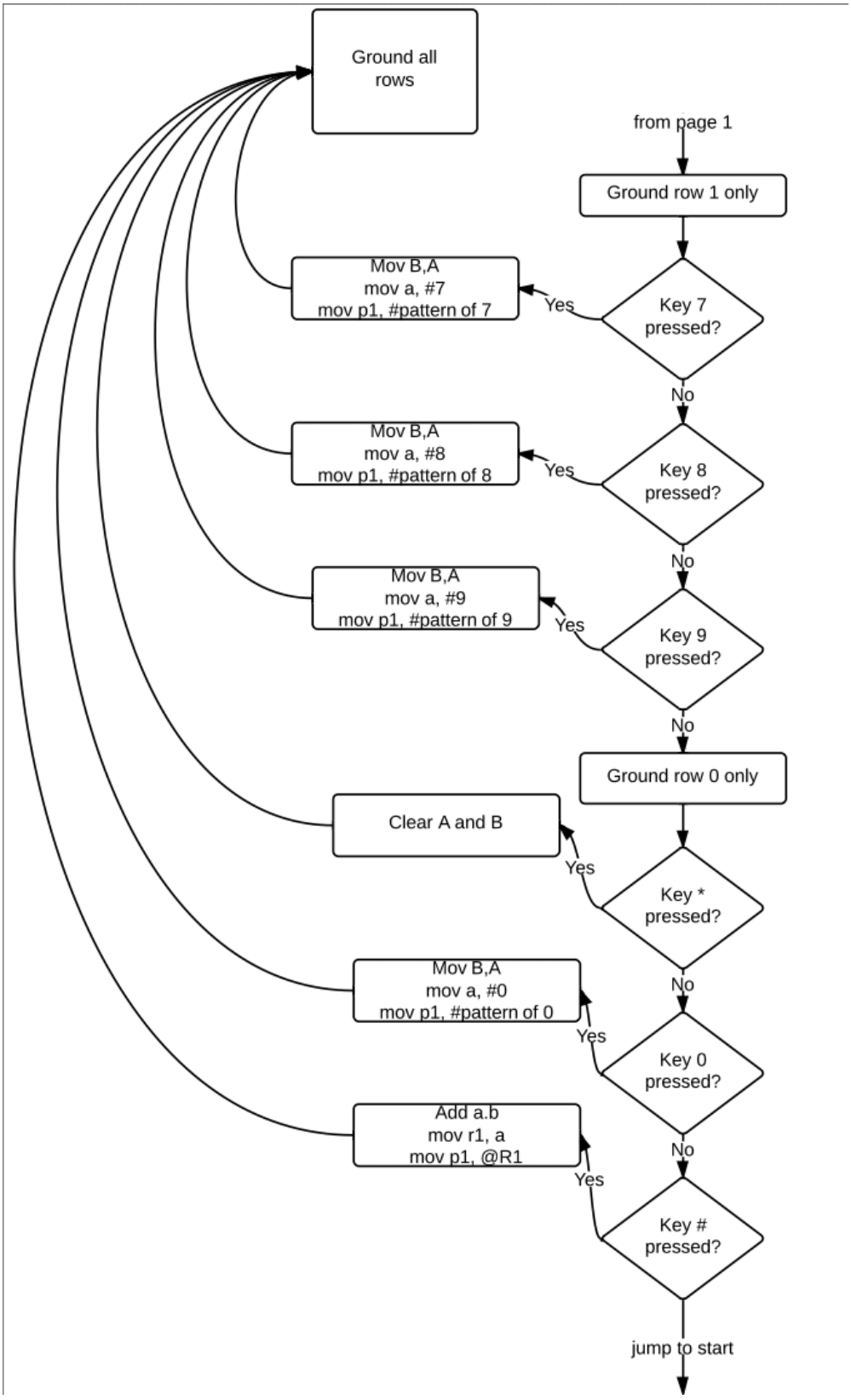
This lab was effective in teaching me how to use the 8051 seven segment display and keypad. I also became familiar with accessing other memory locations using the @Rn code.

Program-

```
Org 30h
;moving the values to be displayed on 7 segment display to data memory
mov 0, #11000000b
mov 1, #11111001b
mov 2, #10100100b
mov 3, #10110000b
mov 4, #10011001b
mov 5, #10010010b
mov 6, #10000010b
mov 7, #11111000b
mov 8, #10000000b
mov 9, #10010000b
;;check key released function
check_key_released:
jnb p0.4, check_key_released
jnb p0.5, check_key_released
jnb p0.6, check_key_released
;;enabling the 1st 7 segment display
clr p3.4
clr p3.3
;;scanning row 1
scan:
        clr p0.3
        setb p0.0
        setb p0.1
        setb p0.2
         jb p0.6, next3         ;1
                                        mov b, a
                                         mov a, #00000001b
                                        mov P1, #11111001b
                                        jmp check_key_released
         next3:     mov b, a ;2
                                        jb p0.5, next4
                                        mov a, #00000010b
                                        mov p1, #10100100b
                                        jmp check_key_released;
         next4: mov b, a ;3
                                        jb p0.4, scan1
                                        mov a, #00000011b
                                        mov p1, #0b0h
                                        jmp check_key_released
;;scanning row 2
scan1:
        clr p0.2
                setb p0.0
                setb p0.1
                setb p0.3
        jb p0.6, next ;4
                        mov a, #00000100b
                        mov p1, #099h
                        mov b, 1
```

```
                        jmp check_key_released
        next:jb p0.5, next1 ;5
                         mov a, #0000101b
                        mov p1, #092h
                        mov b, a
                        jmp check_key_released
        next1:jb p0.4, scan3 ;6
                         mov a, #0000110b
                        mov p1, #082h
                        mov b, a
                        jmp check_key_released
;;scanning row 3
scan3:
        clr p0.1
        setb p0.0
        setb p0.2
        setb p0.3
                jb p0.6, next5 ;7
                                mov a, #0000111b
                                mov p1, #0f8h
                                mov b, a
                                jmp check_key_released
        next5:  jb p0.5, next6 ;8
                                mov a, #0001000b
                                mov p1, #080h
                                mov b, a
                                jmp check_key_released
        next6:  jb p0.4, scan4 ;9
                                mov a, #00010001b
                                mov p1, #090h
                                mov b, a
                                jmp check_key_released
;;scanning row 4
scan4:
                clr p0.0
                setb p0.3
                setb p0.1
                setb p0.2
        jb p0.6, next7
                                ;Clear button
                                mov a, #'*';ASCII code of * to A
                                clr a
                                mov b, #000h
                                mov p1, #0ffh
                                jmp check_key_released
next7:  jb p0.5, next8
                                mov a, #00000000
                                mov p1, #0C0h
                                jmp check_key_released
next8:  jb p0.4, next9
                        ;add button
                        add a, b
                        mov R1, a
                        mov p1, @R1
                        jmp check_key_released
next9: ljmp scan
end
```


ground all rows
no
all keys open?
yes
Ground row 3 only
Mov B,A
mov a, #3
mov p1, #pattern of 3
Yes
Key 3 pressed?
No
Mov B,A
mov a, #2
mov p1, #pattern of 2
Yes
Key 2 pressed?
No
Mov B,A
mov a, #1
mov p1, #pattern of 1
Yes
Key 1 pressed?
No
Ground row 2 only
Mov B,A
mov a, #4
mov p1, #pattern of 4
Yes
Key 4 pressed?
No
Mov B,A
mov a, #5
mov p1, #pattern of 5
Yes
Key 5 pressed?
No
Mov B,A
mov a, #6
mov p1, #pattern of 6
Yes
Key 6 pressed?
No


Ground all rows
from page 1
Ground row 1 only
Key 7 pressed?
Yes
Mov B,A
mov a, #7
mov p1, #pattern of 7
No
Key 8 pressed?
Yes
Mov B,A
mov a, #8
mov p1, #pattern of 8
No
Key 9 pressed?
Yes
Mov B,A
mov a, #9
mov p1, #pattern of 9
No
Ground row 0 only
Key * pressed?
Yes
Clear A and B
No
Key 0 pressed?
Yes
Mov B,A
mov a, #0
mov p1, #pattern of 0
No
Key # pressed?
Yes
Add a.b
mov r1, a
mov p1, @R1
jump to start