EE5811 : FPGA LAB
ASSIGNMENT 4

Himanshu Yadav
IS21MTECH11003

February 10, 2022

Problem

The AVR assembly code for the following expression. Also interface a seven segment display to show the inputs

$$(A + B) . (C + D)$$

Solution

Figure 1 shows a seven segment display with pins a, b, c, d, e, f, g, dot. Each of these pins is connected to a LED. Table 1 shows how to generate the numbers on the display along with what need to be output on the PORTD of ATmega328P to display the same.

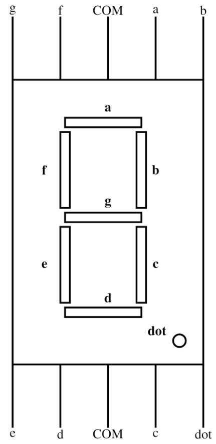

Figure 1: Seven Segment pinout

g	f	e	d	c	b	a	dot	hex	PORTD
0	0	0	0	0	0	1	0	0	0x02
1	0	0	1	1	1	1	0	1	0x9E
0	0	1	0	0	1	0	0	2	0x24
0	0	0	0	1	1	0	0	3	0x0C
1	0	0	1	1	0	0	0	4	0x98
0	1	0	0	1	0	0	0	5	0x48
0	1	0	0	0	0	0	0	6	0x40
0	0	0	1	1	1	1	0	7	0x1E
0	0	0	0	0	0	0	0	8	0x00
0	0	0	0	1	0	0	0	9	0x08
0	0	0	1	0	0	0	0	A	0x10
1	1	0	0	0	0	0	0	B	0xC0
0	1	1	0	0	0	1	0	C	0x62
1	0	0	0	0	1	0	0	D	0x84
0	1	1	0	0	0	0	0	E	0x60
0	1	1	1	0	0	0	0	F	0x70

Table 1: Truth table for CA Seven Segment Display

```
.nolist
.include "m328pdef.inc" ; Define device ATmega328P
.list

ldi r16,low(RAMEND) ; RAMEND address 0x08ff
 out SPL,r16        ; Stack Pointer Low SPL at i/o address 0x3d
 ldi r16,high(RAMEND)
 out SPH,r16

ldi r16,0b11111110
out DDRD,r16           ; pins D1 - D7 are set as output

ldi r16,0b00110000      ; pins D12 - D13 are set as output
out DDRB,r16

ldi r16, 0b00000101    ;  the last 3 bits define the prescaler, 101 => division by 1024
out TCCR0B, r16

ldi r18,0b00000000
out PortD,r18
ldi r25,0x0f

rcall save_value        ; saving required values at address 0x0200-0x020F
ldi r20, 0x10

loop:
dec r20
rcall DISPNUM
brmi save_value

rcall BITSEP

or r21,r22           ; (A+B).(C+D)
or r23,r24
and r21,r23          ; final result in r21
lsl r21
lsl r21
lsl r21
lsl r21
lsl r21

rcall NOR1           ; NOR Equivalent
rcall NOR2
rcall NOR3           ; final result in r11
lsl r11
lsl r11
lsl r11
lsl r11

or r21,r11
out PortB,r21       ; pushing result to pin D12 and D13

ldi r19, $32
rcall DELAY
rjmp loop

DISPNUM:            ; routine for displaying number on seven segment
    ld r17, Z+
    out PortD,r17
    ret
```

```
DELAY:         ;this is delay (function)
               ;times to run the loop = 64 for 1 second delay
    lp2:
        IN r16, TIFR0        ;tifr is timer interupt flag (8 bit timer runs 256 times)
        ldi r17, 0b00000010
        AND r16, r17         ;need second bit
        BREQ DELAY
        OUT TIFR0, r17       ;set tifr flag high
        dec r19
        brne lp2
    ret




BITSEP:                     ; Routine for bit seperation
    mov r21, r25     ;A LSB
    sub r21,r20
    andi r21,0x01
    mov r11,r21

    mov r22, r25     ;B
    sub r22,r20
    andi r22,0x02
    lsr r22
    mov r12,r22

    mov r23, r25     ;C
    sub r23,r20
    andi r23,0x04
    lsr r23
    lsr r23
    mov r13,r23

    mov r24, r25     ;D MSB
    sub r24,r20
    andi r24,0x08
    lsr r24
    lsr r24
    lsr r24
    mov r14,r24

    ret

NOR1:                   ; NOR gate Logic
    or r11,r12
    com r11
    ret

NOR2:
    or r13,r14
    com r13
    ret

NOR3:
    or r11,r13
    com r11
    ret
```

```
save_value:                 ; Routine for saving constant numbers in SRAM
    ldi Zl, 0x00            ; Reset address to 0x0200
    ldi Zh, 0x02

    ldi r19, 0x02
    st Z+, r19      ; store to current address and increment the address
    ldi r19, 0x9e
    st Z+, r19
    ldi r19, 0x24
    st Z+, r19
    ldi r19, 0x0c
    st Z+, r19
    ldi r19, 0x98
    st Z+, r19
    ldi r19, 0x48
    st Z+, r19
    ldi r19, 0x40
    st Z+, r19
    ldi r19, 0x1e
    st Z+, r19
    ldi r19, 0x00
    st Z+, r19
    ldi r19, 0x08
    st Z+, r19
    ldi r19, 0x10
    st Z+, r19
    ldi r19, 0xc0
    st Z+, r19
    ldi r19, 0x62
    st Z+, r19
    ldi r19, 0x84
    st Z+, r19
    ldi r19, 0x60
    st Z+, r19
    ldi r19, 0x70
    st Z+, r19

    ldi Zl, 0x00        ; Reset address to 0x0200
    ldi Zh, 0x02
    ret
```