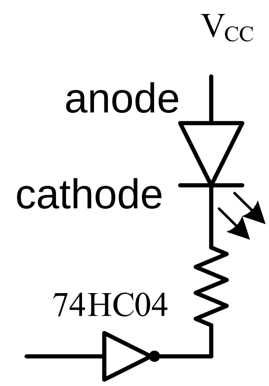
CEG 3136 – Computer Architecture II
Tutorial 4 – The Parallel Ports - Diodes Fall 2019

Part A – Driving LEDs

The adjacent circuit illustrates the circuit used to drive an LED. The 74HC04 NOT gate will bring the resistance down to 0V when a 1 is applied to the gate. Calculate the value of the resistance given a current of 10 mA through the circuit, V_{CC} = 5V, and a voltage of 2V across the LED when the 10mA current flows through the LED.



Using voltage division (Kirchoff's voltage law):

$\mathbf{V_{CC} = V_R + V_{LED}}$

$\mathbf{V_R = V_{CC} - V_{LED} = 5 - 2 = 3V}$

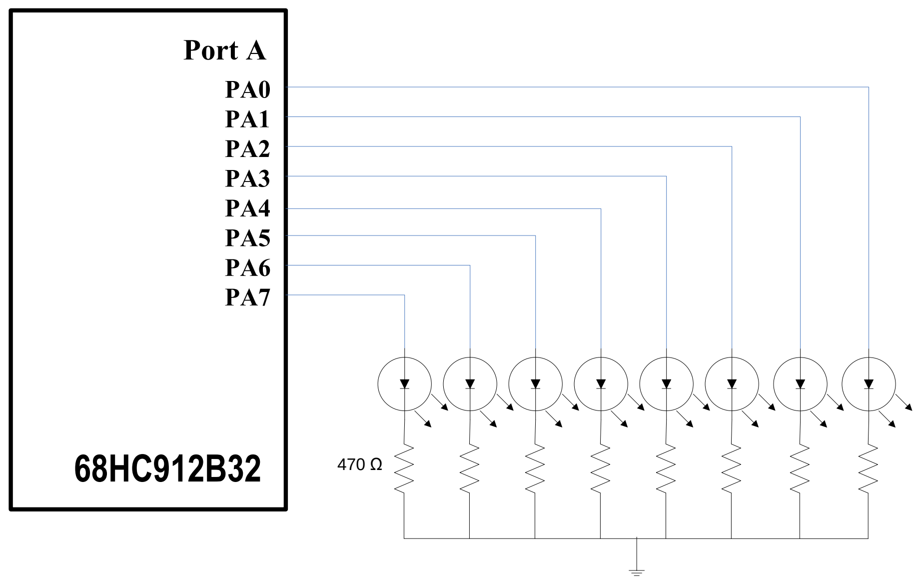
$\mathbf{V_R = R * I_R}$

$\mathbf{R = V_R / I_R = 3V / 10\ mA = 300\ \Omega}$

What is the current in the circuit when a 0 is applied to the gate. What effect does this have on the LED?

0 mA since the gate will generate a voltage of 5V which means that a there is a 0 V drop across the resistor and LED. With a 0 V drop, no current will flow across the diode and thus the LED is turned off.

PART B – Driving LEDs with the MC68HC12 Port A.



Consider the above bank of LEDs connected to Port A.

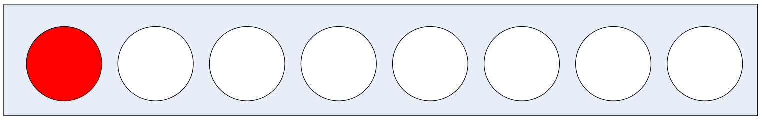
1) Explain how the above LEDs will operate, i.e., what voltage level are applied to the LED anodes to light it up.

To turn the LED on, Port A pins must be configured as output pins. To turn an LED on, 5 volts must be applied at the corresponding pin (a 1). To turn the LED off, 0 V is applied at the corresponding pin (a 0).

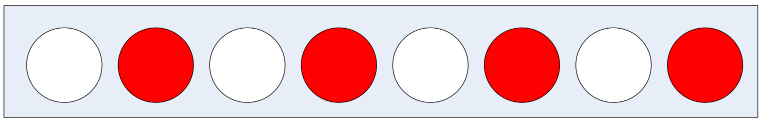
2) Define how to configure Port A to allow the micro controller to light up the LEDs.

To configure Port A pins as output pins, the data direction registers for the port (DDRA at address $02) must be set to $FF. Setting a bit in DDRA sets up the corresponding pin to an output pin.

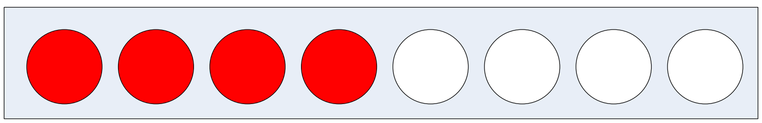
3) What register is used to control the LEDs? **The Port A data register at address \$0.** What hex values are used to provide the following patterns in the bank of LEDs?



\$80 (%1000 0000)



\$55 (%0101 0101)

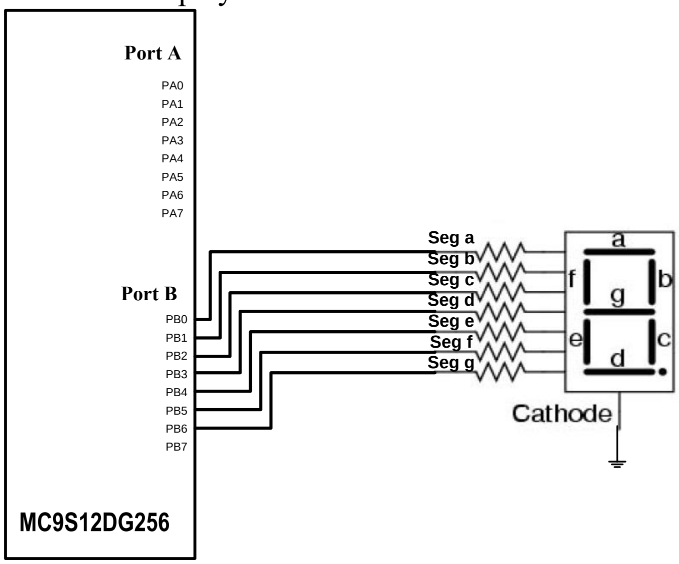


\$F0 (%1111 0000)

4) Given the subroutine “delay(num)” that generates a delay of num milliseconds (num can vary from 0 to 255 and must be placed in accumulator B), develop a program that will run a light across the bank of LEDs, first from left to right and then from right to left giving the impression that the light is bouncing from side to side. Use ¼ second delay to move the light from LED to LED.

```
; Parallel Port A Registers
PORTA EQU $00   ; Port A Data Register
DDRA EQU $02    ; Port A Data Direction Register

; Constants
OUTPUT EQU $FF
COUNT EQU 7     ; Number of moves before reversing
START EQU %10000000 ; Initialization of LEDs
QRTSEC EQU 250  ; 250 milliseconds = 1/4 second
; Code
        ORG $0800
        ; Initialise Port A
        movb #OUTPUT,DDRA
        movb #START,PORTA

Loop    ldaa #COUNT
Loop1   ldab #QRTSEC
        jsr delay
        lsr PORTA  ; move one LED right
        dbne A,Loop1  ; continue to right

        ldaa #COUNT
Loop2   ldab #QRTSEC
        jsr delay
        lsl PORTA  ; move one LED left
        dbne A,Loop2 ; continue to left

        BRA Loop
```

5) Seven segment displays consists of a set of LEDs. Such displays can have either the cathodes or anodes tied together. Below is a diagram that shows a common cathode display. Note that the current limiting resistors are tied to the anodes. To turn on a segment, the cathode must be tied to ground (0V) and 5V applied to the resistor of the corresponding segment anode.
   a. Develop a circuit to display characters on the 7-segment display by connecting it to parallel PORT B of the MC9S12DG256 micro-controller. How would you configure Port B to control the display?



   Port B data direction register: Set to all 1's (i.e. as output pins)
   b. Define a list of bit patterns that are used to display the following: digits 0 to 9, A, b, C, d. Choose some alternate patterns for the characters # and *. Note that all these characters have corresponding keys on the Dragon-12 keypad.

Character	Bit Pattern
0	0011 1111
1	0000 0110
2	0101 1011
3	0100 1111
4	0110 0110
5	0110 1101
6	0111 1101
7	0000 0111
8	0111 1111
9	0110 1111
A	0111 0111
b	0111 1100
C	0011 1001
d	0101 1110
#	0111 0000
*	0100 0110

6) The following circuit illustrates how 4 seven-segment displays are connected to the MC9S12DG256 micro-controller on the Dragon-12 evaluation board. The displays are common cathode displays. Note that the displays' anodes are connected to an 8-bit bus that is linked to Port B and the common cathodes are connected to Port P: DIG3 to PP0, DIG2 to PP1, DIG2 to PP2, DIG0 to PP3. To enable a display, 0V must be applied to the pin on port P connected to the display. A bit pattern written to port B will now have a character appear on the enabled display.

   a. Is it possible to have different characters appear on different displays simultaneously?

      No, since enabling more than one display means that the displays show the same character according to the bit pattern on the bus.

   b. Is it possible to make it look like different characters appear on different displays?

      Yes, by enabling each display for a small period of time with their respective characters. Repeating the refresh of each display at a very high rate will make it look like the characters are being displayed simultaneously.

   c. Discuss possible designs of a display module that would provide the means to display 4 characters (blanks are allowed also) on the 4 displays. Assume you have access to delay subroutines is a Delay module (recall such a module used in Lab 2). Consider that the module should provide the following sub-routines:

      - A subroutine for initialisation;

        Initialises ports B and P (bits 0 to 3) as output pins.

      - A subroutine to update the characters to be displayed for each display;

        The subroutine parameters should define the character to be displayed and the display on which the character should appear
        Two of approaches can be used to save the character: store the character itself or the corresponding bit pattern that can be used to display the character (storing the bit pattern means that conversion of the character is not required when its time to display the character). In either case, a buffer of four bytes are required to store either the char or bit pattern.
        Also, a subroutine to convert the character to a bit pattern is required.

      - A subroutine for updating the displays. The subroutine can retain control for a period of time but must relinquish the control so that the CPU can attend to other tasks. Assume that such control would be given to the module on a regular basis.

        Here is one approach:

          Define a delay for display, say 10 ms.
          Then using the store characters/bit patterns, display each character for 10 ms.
          Repeat the process 3 times, and return from the subroutine.
          This means that the subroutine runs for 120 ms and then relinquishes control back to the calling subroutine.

        For the continuous display on all four displays, this subroutine needs to be called on a regular basis.