ECE 367 - Experiment #8
Synchronous Serial Interface to 7-segment LED Display

Spring 2006 Semester

Introduction

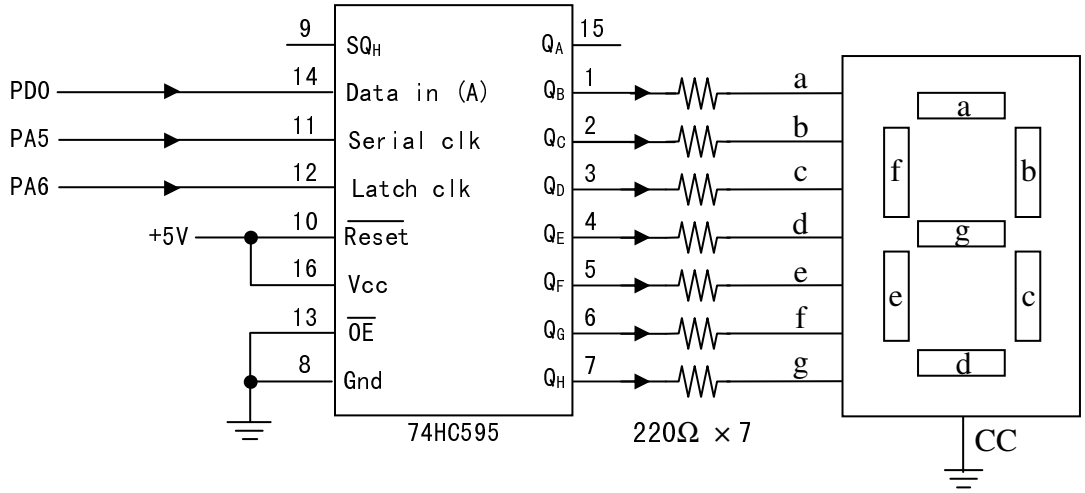
In this experiment you will write code to interface with a seven segment display device in a synchronous serial manner. The goal is to implement a single digit counter, as in Experiment 4, using only three output lines.

Required Hardware

In addition to the MicroStamp11 module, this experiment requires one seven segment LED display, 7 current-limiting (220Ω) resistors in a DIP package, and the serial-in parallel-out shift register 74HC595 IC*.

Wiring Diagram

Build the following circuit on a solderless breadboard:

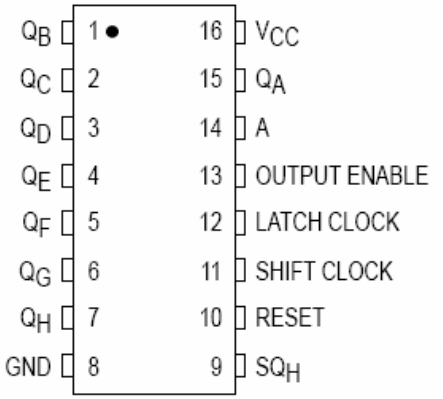


Goal

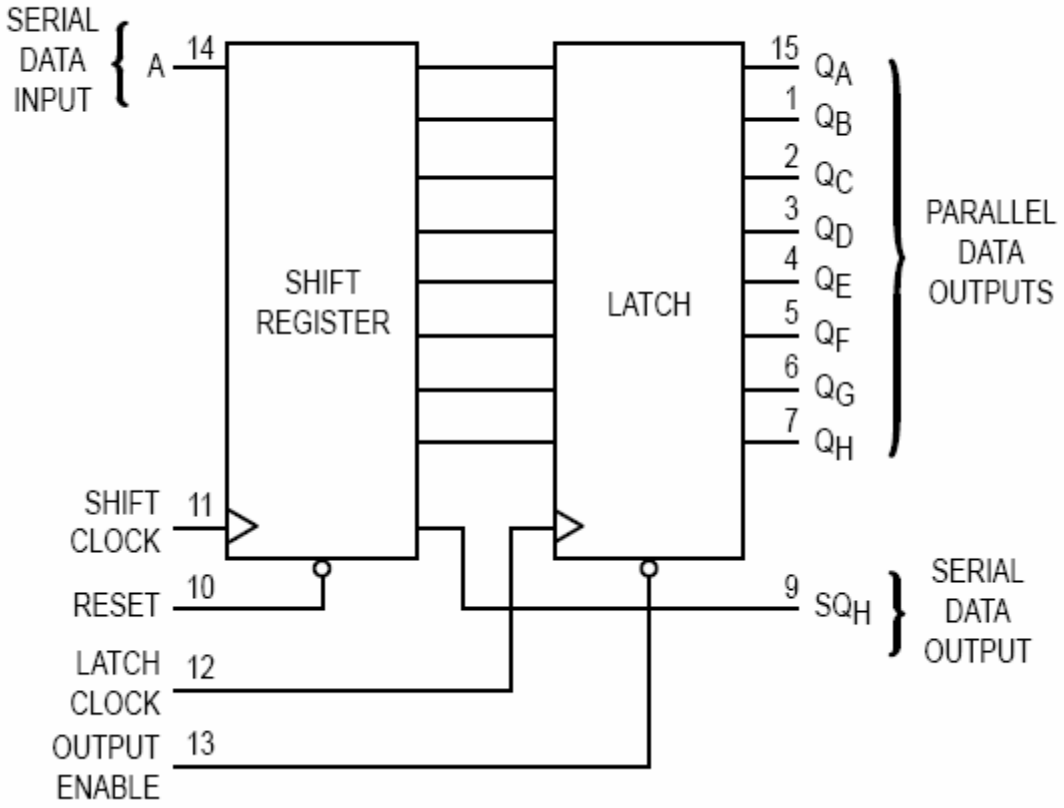
By controlling the output signals generated at PD0, PA5 and PA6, you are to serially transfer data to the seven segment LED display so that it counts up, once per second, as was done in Experiment #4.

*As this is a CMOS integrated circuit, it is susceptible to static discharge damage.

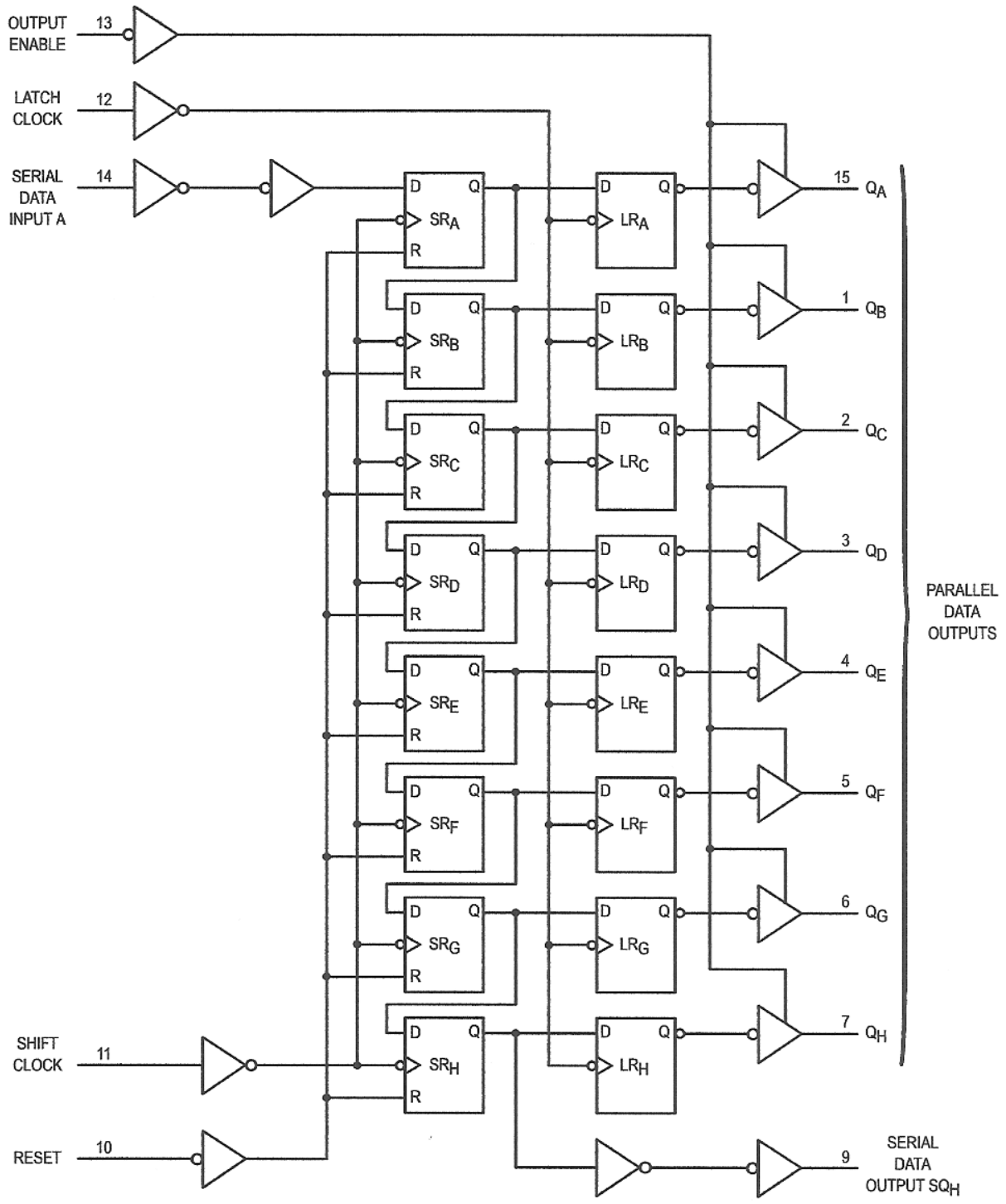
PIN ASSIGNMENT



LOGIC DIAGRAM



EXPANDED LOGIC DIAGRAM



Method

First the data bits are shifted from PD0, one bit at a time, into the shift register. To do so one must output the desired bit to PDO and then pulse the serial clock line (rising edge). Repeat until the desired eight bits have been loaded into the shift register. When finished, pulse the latch clock line (rising edge) to transfer the contents of the serial shift register to the output latch.

Here is a subroutine for transferring a byte of data from microcontroller register ACCA to the serial shift register:

```
Output_a_byte:
          LDAB      #8              ; Initialize bit counter
A0:
          STAA      PortD,X         ; Copy ACCA to PortD (only PD0 is used)
          BCLR      PortA,X,$20     ; Creates low to high transition at PA5
          BSET      PortA,X,$20     ; shifts PD0 value into shift reg
          LSRA                      ; shifts ACCA right by one bit
          DECB                      ; decrement count
          BNE       A0              ; repeat until 8 bits have been transmitted
          BCLR      PortA,X,$40     ; Creates low to high transition at PA6
          BSET      PortA,X,$40     ; shift reg contents sent to output latch
          RTS                       ; Return from subroutine
```