Lab 2: Decoding, Multiplexing, and Sequencing

2.1 Decoder

As you recall, a decoder uses an input address to select one of many possible outputs. Multiplexers use a decoder to select which input line gets sent to the output. The 74154 IC is a 4-bit decoder. That is, it decodes a 4-bit binary number (the 4 *address* inputs) to select one of $2^4 = 16$ possible outputs. Two pages of the 74154 manufacturer's data sheets are attached, for reference.

Construct the circuit shown in Fig. 1. The 7493 is a 4-bit binary counter. (The data sheet is attached.) We will learn more about counters later. But its basic function is simple: It counts the number of pulses input on pin 14. The 4 binary digits representing this count are output on pins 12, 9, 8, and 11, as shown in Fig. 1. We are using these 4 bits of counter output simply to set up different adresses for the decoder. Use a debounced switch on your prototype board to generate the counter input pulses. Check that the counting is proceeding correctly using the 7-segment display, as shown. Toggle through all address inputs to the 74154 and measure the state of the 16 outputs. The outputs can be measured using a scope, logic probe, DVM, or the LEDs on your prototype board. Hence, determine and record the truth table for the 74154. Does it use "positive true" or "negative true" logic ?

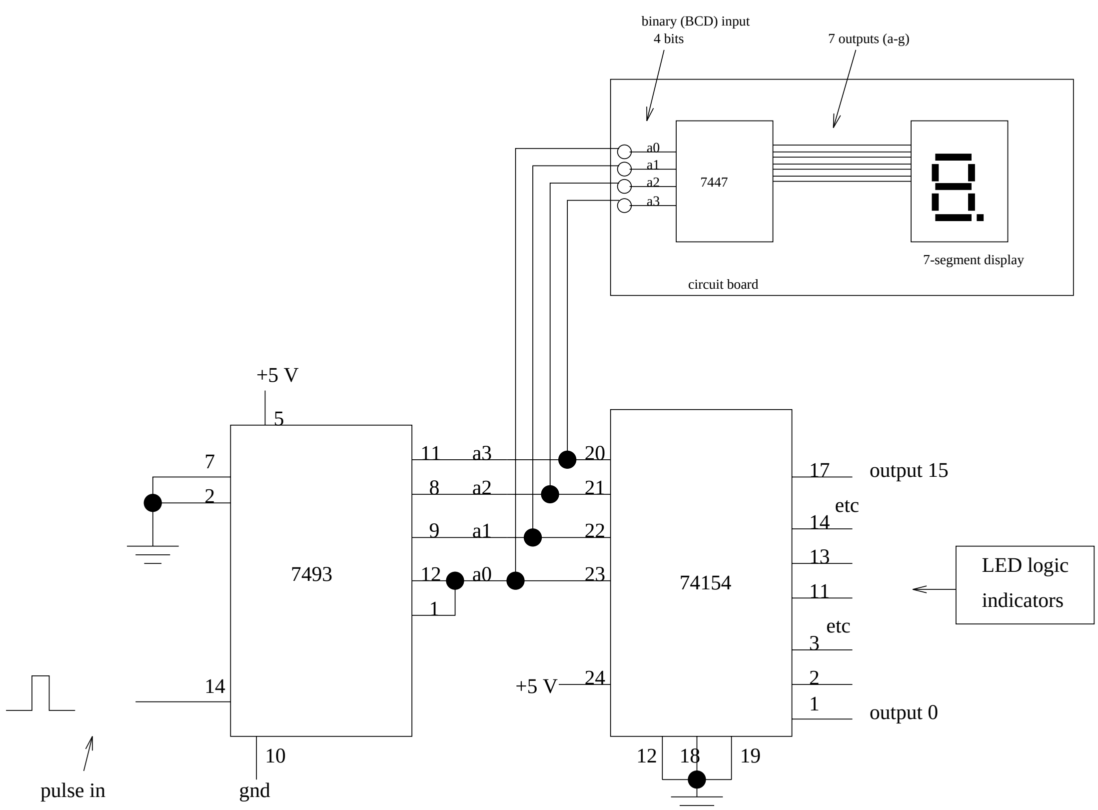

Figure 1: Decoder circuit schematic. Note that pin 12 of the 74154 is not an output.

2.2 Sequencer

A sequencer is a decoder for which the input is incremented sequentially by a counter. The sequence can continuously repeated by resetting the counter. Hence, this is functionally the same as the previous circuit, except now we will include a "programmable" `reset` function. In addition, we will change from a binary (*i.e.* unsigned binary) counter and decoder to a BCD counter (7490) and decimal decoder (7442). This is shown in Fig. 2. The pin assignments are similar, but not identical, to the previous circuit.

Any one of the decoder outputs can be inverted and sent to the counter `reset` (pin 2), as shown. If, for example, output channel $n = 5$ (of the channels 0–9) is connected to the counter `reset`, what will be the highest count that you see on the 7-segment display? Why is the 7404 inverter needed?

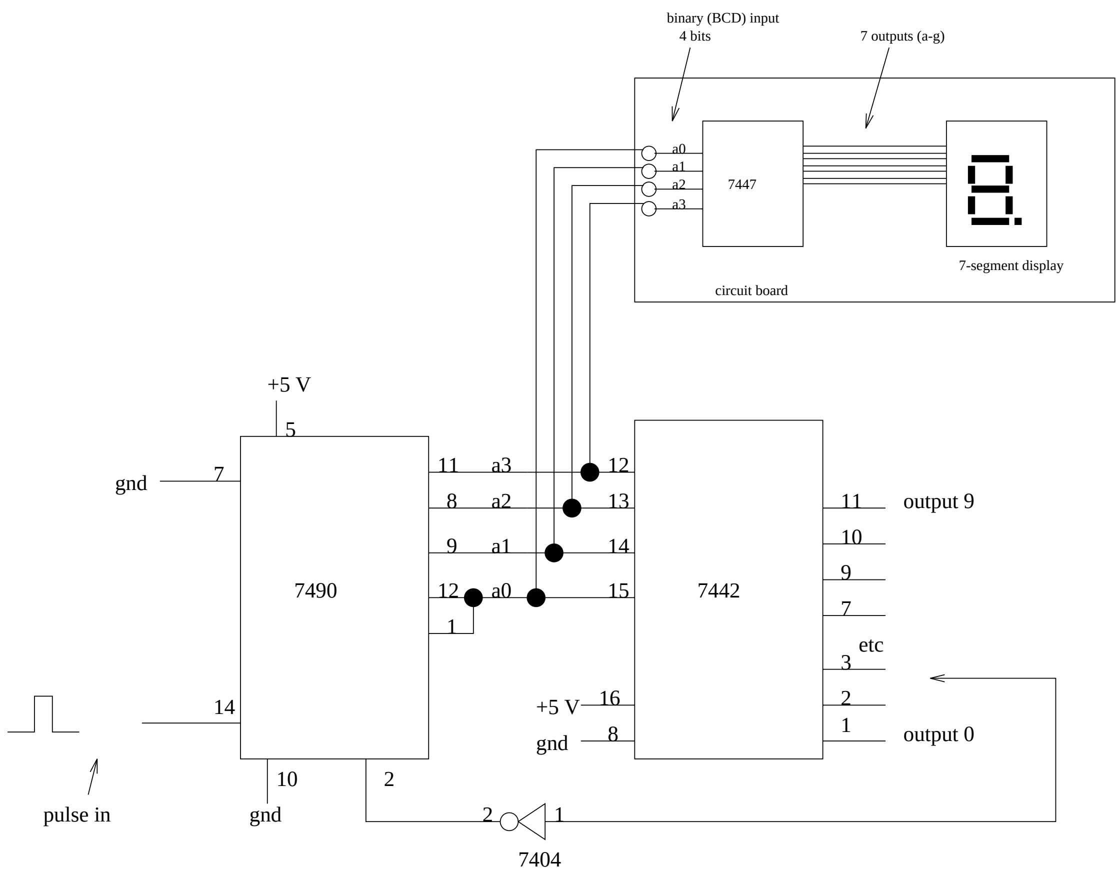

Figure 2: Sequencer circuit schematic. Note that pin 8 of the 7442 is not an output. Be sure to connect the reset of the 7490 (pin 2) via a 7404 inverter, as shown.

2.2.1

Now connect up the circuit of Fig. 2. Does it work as expected? Vary the position of the connection to the `reset` and determine the maximum and minimum number of channels which can be sequenced.

2.3 Demultiplexer

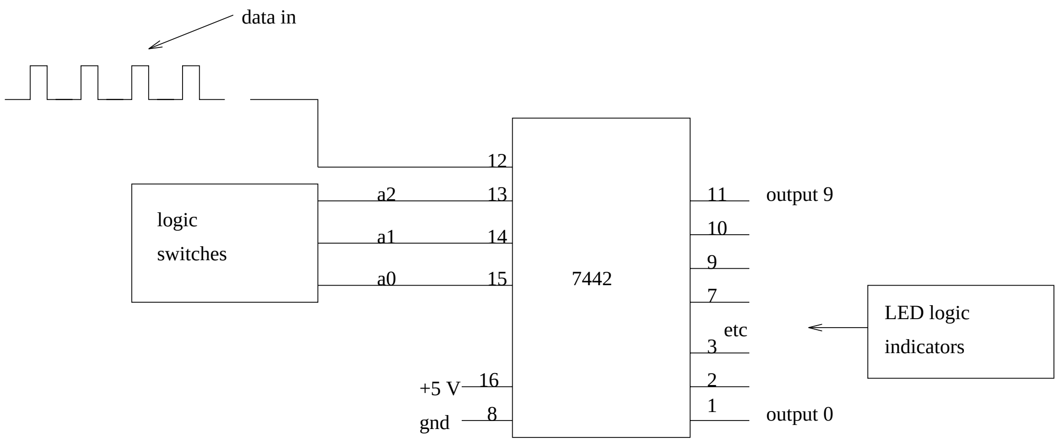

Figure 3: Demultiplexer circuit schematic.

Consider Fig. 3. While a decoder is often used as part of a multiplexer, in this configuration the 7442 decoder becomes a $1 \rightarrow 8$ demultiplexer. Recall that a demultiplexer ("DEMUX") routes data on a single input to one of several possible output locations, as selected by an input address. In this case, the three least-significant address bit inputs are used to select the output location, while the most-significant bit (MSB) at pin 12 becomes the data input. The data gets routed to output lines 0 to 7. To see how this works, examine the 7442 truth table shown below in Table 2. It uses "negative true" logic for the outputs, but in the table (from the manufacturer) 0 =`LOW` and 1 =`HIGH`. We see that when the MSB ("D") is `LOW`, the output selected by the other three bits is `LOW`, and vice versa.

Table 2 Truth table for 7442

INPUTS				OUTPUTS									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Construct the circuit shown in Fig. 3. Use a slow ($\sim$ 1 Hz) TTL square wave as the data input. Select an output channel using logic switches. Connect the outputs 0–7 to the 8 logic LEDs on your prototype board. Does the input data appear at the selected output? What appears at the other channels? Repeat with a different channel.

2.4 Multiplexer-demultiplexer

Here we will construct the circuit of Fig. 4, which consists of a multiplexer (74150) which sends data to a demultiplexer (74154) over a single line. Hence the two devices communicate via a *serial data link*. Sending data over such a serial link clearly saves making many connections, which is of practical importance. The MUX and DEMUX use the 4-bit address bus shown in Fig. 4 to communicate which data channel is active. In your circuit, the address code will be selected using a counter (7493), as was done earlier. Again, the 7-segment display is used to display this address.

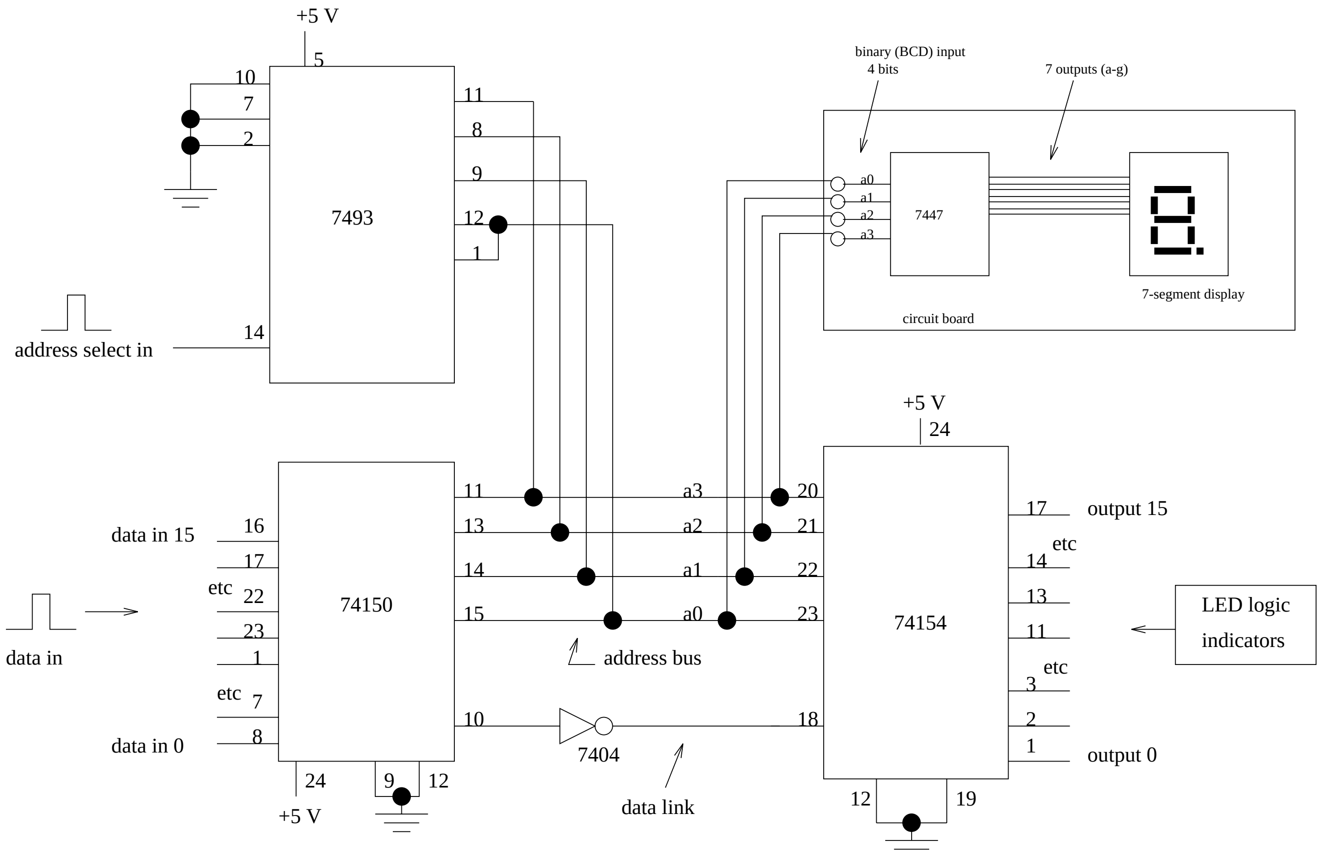

Figure 4: Schematic for serial data transmission via MUX/DEMUX.

Connect the circuit shown in Fig. 4. The data to be transmitted ("data in") can again be a slow TTL pulse train. And a debounced button input to the 7493 counter can again be used to select the address. The inverter is used to keep input and output data in phase. Increment the counter to select output channel 5. Which data pin(s) on the 74150 (data in) and 74154 (data out) have been selected? Verify this with a logic probe or LED. Now input data to this active 74150 pin. Does the input data get faithfully transmitted to the correct output? Move the data input to a different MUX pin. Now select this channel. Does data now appear at the 74154? At which channel?