

Part I

0. In this part of the lab you investigate the '164 a serial-in, 8-bit-parallel-out, shift register.
1. Press in (near the LEDs) a '164. Power the chip “at the corners.” Take CLOCK and CLEAR to separate NO push buttons. Take A and B to data switches. Take the outputs Q_A – Q_H to LEDs.
2. Read the spec sheet, paying particular attention to the Function Table and the output traces. Test out your chip by switching A and B to high, and clocking. You should see your LEDs sequentially light. What sort of edge-triggered clock does the '164 use? (Report the observations that led to your conclusion!) Is the CLEAR synchronous? (Provide evidence!) How do the serial inputs A and B interact?

Part II

0. In this part you will construct the MonoBus (MB). The MB is a useless device that demonstrates several interesting techniques. First the MB is a Tri-State bus. This means that the disastrous consequences of wiring together the *outputs* of different gates are avoided by designing the circuit so that only one '125 gate is out of its Hi-Z state at a time. That one “talking” gate determines the logic level of the bus as the other outputs connected to the bus are in a Hi-Z (essentially disconnected) state; the receiver will “hear” only the one talking gate. Second, the MB is a serial bus. The 4-bit nibble transmitted over the bus is sent bit-by-bit, LSB first, on a single DATA wire. Note closely how parallel data in the '143 is “serialized” and how serial-to-parallel conversion is done by the receiver. If the LSB is sent first, *where on the '164 is it when it is grabbed by the '175?* Third, the MB—like all buses—uses control lines to organize the transfer of data over the bus. One MB control line (Bit Valid, BV) indicates with a positive-going edge that the bit on the bus should be valid now. Another MB control line (End of Transmission, EOT) indicates with a positive-going edge that the MSB has just been transmitted. (The LSB of the next nibble will follow immediately.) Note that four wires (DATA, BV, EOT, and ground) constitute the MB.

The '143 counter/latch/decoder/driver is used here with a push-button clock simply to quickly create BCD nibbles to be transmitted. The '163 binary counter is used to cycle through the 4 “devices” (bits really), allowing each to sequentially control the bus. The '138 is used to decode the address of the device named by the '163 and enable only that gate on the '125 (new chip: tri-state buffer). On the receiving end, the MB DATA is parallelized by a '164, grabbed by a '175 (new chip: edge-triggered D FF), and finally decoded for display by a '48.

1. Examine the block diagram on the following page. Understand how the MB works. Cut and tape the below timing diagram into your notebook. Fill in the empty graphs for the case of the to-be-transmitted nibble $9 = 1001_2$. Label your graph precisely showing the instant when (a) each bit of data first appears on the MB (e.g., when is the LSB on MB), (b) when each bit of data is grabbed by the shift register, and (c) when the D FF grabs the nibble from the shift register. Also report which output of the '164 has the MSB and which has the LSB when the D FF grabs the nibble. Explain the purpose of the inverter on BV (i.e., what problem would you foresee if it is removed?)
2. Build the MB transmitter on one side of the board, and the MB receiver on the other. Think about how to build the project so it can be checked-out sequentially: it's hard to debug a multi-chip circuit unless you're sure each part is working correctly.
3. Connect the transmitter and receiver and show the working example to your instructor.
4. Remove the inverter on BV. What happens? Can you explain it?

