(A) Digital data arrives serially. It is synchronous. That is, arriving bits are available on the wire for a clock period, and are valid at a rising clock edge. However, new data does not always arrive on every clock period, so a separate signal, incomingData, is made TRUE whenever a bit of data is available for input.

Design a circuit that loads a bit of incoming data into a buffer. When data is required, (as signalled by a control input called outputData) the earliest item in the buffer should be output.

The buffer should have storage for 4 bits of data. It should stop accepting data when it is full.

(B) Design a circuit that loads serial data into a 4-bit shift register when a control signal, accepting, is TRUE. As in the circuit described above, incoming data is synchronous, but not continuous, and an incomingData signal is used by this circuit also.

(C) A large seismic event has been observed on the moon. It appears to have come from the far side. And a circumlunar expedition has been hurriedly equipped and sent to investigate.

Having arrived at the suspected site of the seismic event, the expedition has performed a number of tests, and has detected a faint signal that appears to be coming from the interior of the moon. It’s a high-speed sequence of 4-bit binary numbers. They’re arriving via a combination of directed magnetic pulses that a member of the team has converted into conventional digital signals.

It seems likely that communication with the originators of the signal will be possible, but the signals are growing steadily fainter at such a rate that they will disappear entirely within a week. If two-way communication is to be established before the suspected sublunar inhabitants disappear entirely, wiped out by the moonquake, the code must be analysed as a matter of the highest priority.

This is a major problem, for two reasons. First, communication from inside the moon was unexpected, so the expedition was not equipped to cope with it. Secondly, the seismic site is on the far side of the moon, and the expedition was set up in a hurry after the seismic collapse occurred; there was no time to organize communications satellites to enable it to remain in contact with the earth. It would take more than a week just to get back to a location from which communication with the earth is possible. There’s obviously no time to wait till the computing facilities available on earth can work on the problem. The expedition itself will have to do the job.

Fortunately, the expedition includes some members who have expertise in linguistic analysis. Their techniques involve extensive computer analysis. There is some spare computing capacity; a few old PCs that were used for other tasks at the start of the expedition and will be used again when the expedition regains contact with the earth. Unfortunately, because of the specialized functions which those particular PCs perform, they lack standard serial input ports. However, each is equipped with a low-speed parallel input port capable of accepting four-bit inputs.

You have therefore been asked to design an input buffer that will allow PCs to pick up data from the incoming stream. The linguistic analysis works on eight-nybble (four-bit units of data are sometimes called nybbles, ostensibly because a nibble is a small bite) chunks of data, so the buffer is required to store eight 4-bit words of incoming serial data. When a request for data occurs, it should parallel-output the 4-bit word that was first stored in the buffer. When it is full, it should stop loading data, and pass the data and the incomingData signal through to an identical downstream buffer.

Note that output from the buffer is not coordinated with input. Thus a PC can request a word from the buffer just after the buffer has started forwarding an incoming word to the next buffer. This word should continue to be forwarded, so that it is not split between two PCs.