scribes a simple bus protocol and then uses it to illustrate how sequences can be used to check logic models of the protocol.

9.3.1 The SimpleBus Definition

The SimpleBus protocol is shown in Figure 9.5. It connects a processor and memory across a bus with a simple protocol. This same example was used in Chapter 6.3. The processor exclusively drives the values address, start, and read. The values data and dataValid are shared bus lines that can be driven by either the processor or the memory. If it's a read, the memory drives the data lines with the value read from memory and it sets dataValid to indicate to the processor that the value is on the data lines. If it's a write, the processor drives the data lines with the value to be written into the memory. It sets dataValid to indicate to the memory that the values are on the data lines. The address bus is multiplexed with the top 8 bits of the address being sent first followed by the lower 8 bits.

A read operation starts in state S1 with the processor driving the upper address on the address bus and asserting start. In state S2, it drives the lower address on the address bus and asserts read if this is to be a read of the memory. If it's a write, then read is not asserted. The memory is initially watching for start to be asserted and when it sees the start signal, it loads the upper address from the address bus lines. In the next state (S2) it automatically reads the lower address from the address bus lines. It also sees the status of read.

Since Figure 9.5 is showing a read bus cycle, dataValid is changed from z to 0 by the memory in state S3. This is because it will be the signal for the processor to read the data lines which will be driven by the memory. This is shown happening in state S5. Note that state S4 is a place holder for many states while the processor waits for the memory to read the requested value and place it on the data lines (or where the memory waits for the processor to send the value to be written). In state S6, the data and dataValid lines become not-asserted again; the next bus cycle could start in this state with the assertion of the address lines and the start signal.

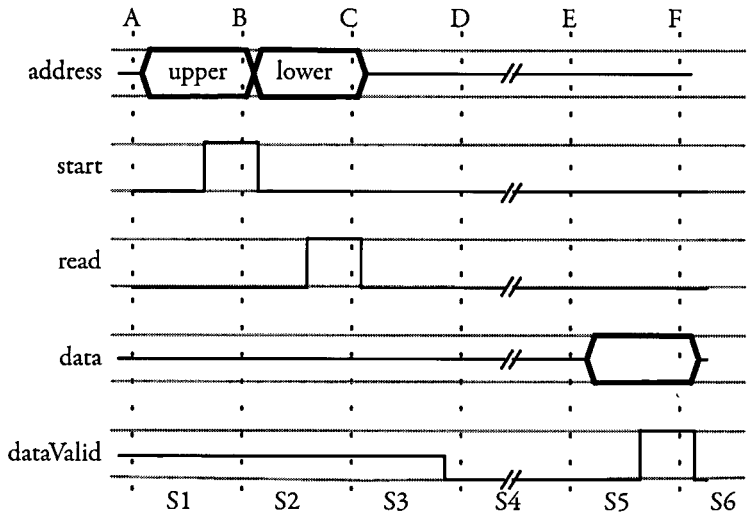

Figure 9.5 — SimpleBus Protocol Timing Diagram

The signal lines for a bus write have only one difference: read is not asserted by the processor. In the case of a write, all the rest of the lines are the same. The distinction is that in a write bus cycle, the processor drives the data and dataValid lines and the memory uses these to load the value on the data into the memory.

Figure 9.5 also shows many of the waveform details that are not specifically stated here. For instance, start and read are only asserted for one state time; otherwise they are not-asserted. Or that the earliest time that start can appear is in the state after dataValid (which would be S6 here).