Multiplexor bus and local address event bus protocol

Steve Deiss Rodney Douglas Mike Fischer Misha Mahowald
Tony Matthews
Applied Neurodynamics
MRC Anatomical Neuropharmacology Unit, Oxford OX1 3TH, England
Research Machines

April 22, 1994

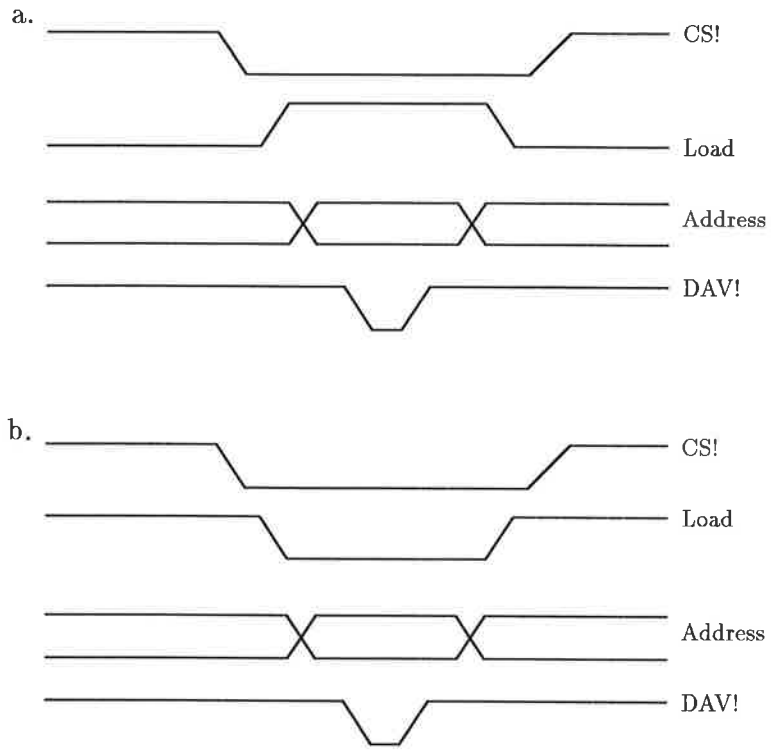

Figure 0.1: The input sequence is initiated by selecting this particular multineuron chip with its !Chip Select line. Next, the type of interaction is specified by the LOAD line assuming the appropriate state. The LOAD line is high when a parameter will be set (a) and the LOAD line is low when this address should be interpreted as a presynaptic event (b). The address on the bus is established. After a pre-defined settling time, the !DAV line goes low, indicating that the address should be read.

The multiplexor bus is written on by the DSP (indirectly) and only listened to by the Multineuron Chips (MNCs). Therefore, all of the signals are driven by the DSP. The analog inputs subserve four conceptually different functions: biophysical parameters, synaptic weights, instrument enabling, and chip identification. In fact, there are only two kinds of cycles on the multiplexor bus, as illustrated in Figure 0.1. Translated presynaptic address event are supplied in one mode, and all of the other functions are supplied in the other.

The local address event bus potentially accommodates multiple senders and receivers. The coordination between thes multiple elements is maintained by a bus arbiter. The signal protocol is illustrated in Figure 0.2. Senders with data assert a request. One of the senders is acknowledged by the bus arbiter. When the request is acknowledged, the sender puts its data on the common local address event bus. In principle, the sender will assert data valid when the data is stable on the bus. We are planning to have the data valid line constructed as a wire OR with a pullup resistor on the test board, so that any sender can pull it down. The present MNC does not have the facility to assert data valid, and this function will have to be performed by the bus arbiter. The data valid pulse is of a fixed width long enough to guarantee that all of the receivers have had time to latch the data.

There are additional signals in Figure 0.2 that will not be implemented in the test board, the Wait! and Timeout!. When implemented, the Wait! signal can be asserted by any receiver upon receipt of data. Wait! prevents the bus arbiter from acknowledging

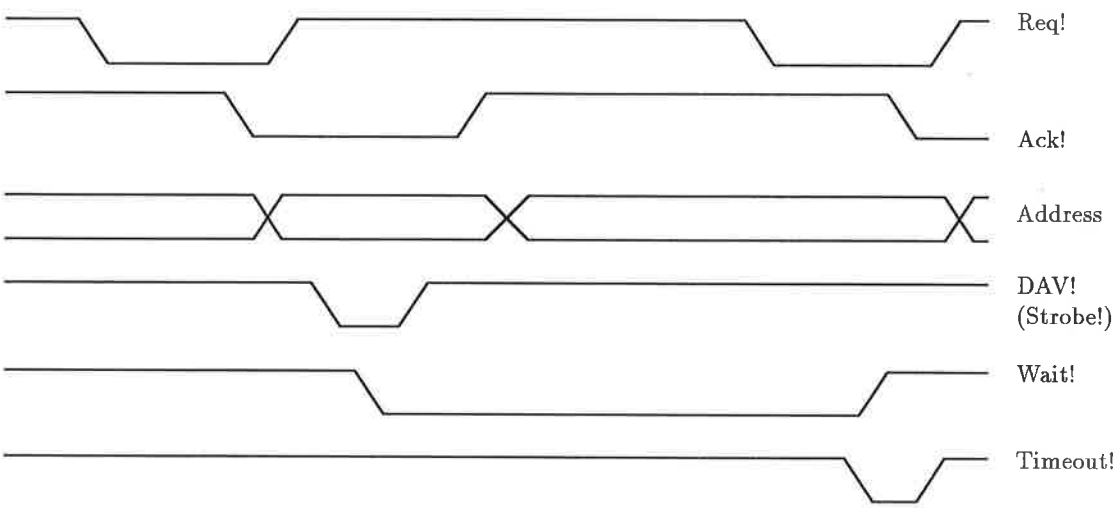

Figure 0.2: Signals on the local address event bus (LAEB).

another request, thereby allowing the receiver time to process the last piece of data. Timeout! is asserted by the bus arbiter if the wait state has lasted longer than a maximum duration. Receivers must remove their Wait! signals whenever there is a timeout.