

Call Sequence

This sequence is used for polling the nodes for interrupt. A node needing service will respond to this sequence. What constitutes a node needing service? It is a node that has been functioned and the function or operation could not be completed immediately. The call sequence is looking for the completion of the operation. Again let's take the example of the disk controller node being functioned to read the drives disk cylinder register. node being functioned to read the drives disk cylinder register. This function takes much time to complete thus the channel loop was freed up immediately after the function was initiated allowing the foreground to function other nodes. (See function sequence write up) When the cylinder register value is finally entered into the node, the node can now respond to a channel call sequence.

A channel call sequence starts when a 54-57 instruction is issued. The channel busy flag sets and a call pulse is sent out on the channel loop with the data lines zeroed out. All the nodes will pass along the pulse. The node(s) needing service will put it's interrupt address on the data lines of the channel loop. The last node in the loop has highest priority. The foreground will receive the call pulse and latch up two parcels of data (the last parcel contains the interrupt address). The call pulse will clear the channel busy flag. The interrupt address will be used by the foreground as a branch address for instruction memory. If the interrupt address value is all zeroes then there were no nodes needing service and a call sequence will have to be performed again.

Data Sequence

The data sequence is used to transfer data from one node to another. The receiving node has to be functioned first to accept data. This node will immediately respond to the function with a response pulse. This free's the channel loop. The transmitting node is functioned next to send data. The transmitting node sends a data pulse with 256₁₀ consecutive parels of data on the channel loop. The receiving node will accept the data pulse and the data. When the receiving node can accept more data it will send a data pulse on the channel loop back to the transmitting node. If the transmitting node wants to send more data then repeat the sequence described above. When all the data has been transferred, the transmitting node upon receiving the data pulse will send a response pulse back to the foreground processor which clears the channel busy flag. Sequence is complete.