from the serial interrupts during the present packet period. As each new sample arrives at the serial input buffer an interrupt is generated and within the interrupt service routine a counter is decremented by one. During each packet period the counter is decremented from its initial value of 70 down to zero, at which point the counter is reset to 70 again.

Associated with each channel is a unique sample number used as a packet send "trigger". Within each jump to the serial interrupt service routine the sample counter is decremented and tested to see if it equals the trigger number. If it does, the node first switches configuration to connect its output FIFO to the NPI bus. Then the node sends packet N-2, already loaded in the output FIFO during the previous packet period, to the mix nodes which expect it. In the case of 60 channels, the trigger numbers range from 68 for channel one down to 9 for channel 60. Channel K sends its packet when the sample counter reaches 69-K. The amount of time required to switch configuration and send a packet is significantly less than the 22.675ms between the arrival of the present sample and the next sample. Thus, no interrupts are undetected and hence each channel node sample counter is always accurate. Furthermore, since the samples from all channels arrive at the serial ports of the respective channel nodes at the same time, all channel nodes have identical counter states. Thus, since each channel node completes its packet send within one uniquely assigned packet period, no packet collisions can ever occur and the channel number order of the packets received at the mix nodes is determined.

Mix nodes also keep globally synchronized sample count but in a different manner. The DSP32C features a simple serial DMA capability. With serial DMA enabled, two registers are dedicated to use as serial input pointer and serial output pointer. When a sample is either taken from the output pointer location and placed in the output buffer or arrives at the input buffer and is loaded into the input pointer location, the appropriate pointer register is post-incremented. The range of addresses the pointer takes corresponds to the sample counter values in the channel nodes. For example, at sample counter 70, the pointer is located at some start address A, and when the sample counter decreases to zero the pointer has advanced incrementally to address A + 70*4 (all internal samples are four byte floats). Thus, the location of the pointer can be used to determine what sample count is in the channel nodes. This correspondence between pointer address and sample counter allows the pointer register to be tested against specific addresses in order to trigger mix events which are synchronized with channel node events. In particular, the final "barrier" address (A + 70*4) is used to test if a new packet period has begun and the pointer needs to be reset.

Two further aspects of system timing need to be discussed. First, there is an initial synchronization to ensure that all nodes, both channel and mix, receive the same first "wave" of samples which trigger counter decrements and pointer increments. A random unsynchronized start might occur just at the instant a wave of samples arrived at the serial ports. Minor time skew due to path length differences might cause some nodes to sense a sample arrival while others might not. Those nodes which did not detect a sample arrival will have missed one interrupt or DMA trigger relative to those that did detect a sample arrival, and hence all local counters would not be synchronized. The initial synchronization is achieved by the remote host signalling node number one by writing to a special memory location in node one's local memory. After node one is signaled it waits for a sample to arrive. After the sample arrives (condition input buffer full is set) node one broadcasts a header-only packet to all nodes which causes them to turn on serial interrupts in the case of channel nodes, or turn on serial DMA in the case of mix nodes. After broadcasting, node one also enables its serial interrupt. These actions are achieved in well under the 22.675ms sample time, i.e., well before the next wave of samples. Thus, all nodes are listening for the same approaching wave of samples so that their local counters or pointers are globally synchronized. Finally, a point must be made concerning configuration (Figure 4). A node in the linear array can be configured either with input FIFO enabled to allow packet reception or packet pass-through (configuration one) or with output FIFO connected to allow packet send (configuration two). An attempt to send a packet to or through a node in configuration two, i.e., configured with output FIFO connected, will result in failure. Thus care must be taken to ensure that the path to the target mix nodes consists of nodes all in configuration one. The following method ensures that each node sends its packet only when all nodes between it and its target are in configuration one. First, upon counter reset to 70, all nodes are in configuration one. The linear array is implemented so that the mix nodes are all "to the right" of all channel nodes, and channel node one is furthest "to the left" in the array (Figure 5). At counter time 68, channel node one switches to configuration two and sends its packet. Then at counter time 67, channel node two switches to configuration two and sends its packet, and so on. In each case, all nodes "to the right" of each sending channel node are in configuration one, and all nodes "to the left" are in configuration two. Thus no blocking can occur so all packet sends are successful. Finally, at sample counter 8, all