unscramble in a satisfactory manner. The best way to get IEI and IEO straight is to, first, read and understand the bit in the PIO manual about the 'daisy chain'. There is a good applications note titled "The Z80 Family Program Interrupt Structure" available from Zilog, which explains everthing. Then cut the tracks between 19 - 19, 20 - 20 between each card on the bus. Decide which end of the bus is going to have the highest priority and connect a 10K resistor to bus line 19 at that end. Connect line 20 of the highest to 19 of the next highest, connect 20 of this to 19 of the next, and so on. Whilst you are at it, repeat the process for BAI/BAO, as these are a form of 'daisy chain' also. The cards may then be inserted from the end which has the highest priority, the 'chain' linking through the RAM cards etc. Do not leave gaps in inserting the cards, as the 'chain' would then be broken. A simple test program was written, and each device on the card tested and found to function correctly.

The I/O card allows considerable addressing flexibility, with the 'daisy chain' fully implemented even when devices are left off the board, so that two I/O cards could be used, each only containing two PIOs for instance. On the card itself, the priorities for the 'daisy chain' are fixed, with the CTC being the highest, followed by PIOs 1 - 3. The UART is not implemented within the priority structure. Cascading of two I/O boards is possible, but more may put timing limitations on the priority structure. The Zilog application note explains what could happen.

In general, the I/O card is another carefully thought out product, let down (as usual) by poor documentation. Although somewhat of a specialist nature, it is a must for those who want to speak to the outside world. Of course, interrupts may be a problem, as NAS-SYS 1 is not interruptable, (and there are limitations on the NASBUGs as well). There is a new monitor on the stocks, which overcomes this problem, and has other enhancements. But for most people, interrupts can be handled with a little careful programming. If you are clever enough to work out how to use the interrupts in the first place, avoiding trouble with the monitor should only be a minor inconvenience.

Lastly, a brief description of the CTC, as this device will be unfamiliar to most users. This is a simple but very versatile chip consisting of four virtually identical chanels. At the heart of each chanel is a 256 (8 bit) countdown counter, with a parallel register which contains a preprogrammed constant. At any time the processor can access the countdown counter to determine the count, or preload the constant register. Two inputs are provided for the countdown counter, and may be selected by the appropriate control word. One from the system clock via a divide 16 or divide 256 prescaler, or from an external input. Control words select the sense of the trigger counting edge, or in the timer mode, whether the count will start immediately or after the input of a preselectable input edge. The scheme is as follows:

Preload the constant, allow the countdown to proceed to zero, whereupon the CTC outputs a pulse, and optionally causes an interrupt (telling the CPU which of the 4 counters it was). At zero count, the constant is reloaded into the counter and it all starts again. All four counters work independantly, and may be timers or counters, they may be cascaded allowing very large division ratios to be achieved if required. It really is a clever little beast with dozens of uses.

Nice one Nascom!
