

## MIDAS

By Rick Lugg

I started with NCR in 1966 working as a computer operator in the test and demonstration area of London's Head Office in Marylebone Road. NCR had started an arrangement with Elliott Automation to market their range of 803 then 4100 computers as a mid-sized system for smaller businesses than could justify the 304 and 315.

Shortly after I started, I met another newbie (we were not called that in those days) John Sandison. We ran compilations and tests for NCR Software developers who were building utilities, compilers and applications, as well as some customer work before their systems were delivered.



NCR 500

After a few months I was offered a position in one of the sales districts (#164) which covered the City of London. Our role was to deliver applications for the NCR 500. When an NCR 500 was sold, the customer was provided two applications as part of the deal. In District 164 we saw the need for coding standards for this clever machine and produced a memory template and a series of utility applications to help document (print out) applications to that standard as well as store them on paper tape, punched cards, or the magnetic ledger cards that were a key feature of the system.

We often parlayed the “two applications” into a role where we would provide this package of utilities and assistance to the customers’ programmers to develop the applications themselves, thereby ensuring that the customer could get programs updated, modified and (sometimes) fixed internally.

When I started writing and preparing programs for the 500 I found an odd situation. There was a bank of 8 or 10 paper tape teletypes for the 4100 that often sat unused, whereas there were only two cash register like machines that could spit out 500 paper tape. So amongst the various programmers, demonstrators and other staff it was often quite a wait to produce tape to run a test.

John was still in the 4100 group that was being wound down as the Century was being introduced. We discussed (probably in the pub) how it would be nice to create a converter to take the 8 channel 4100 tape and convert it to 5 channel 500 tape. As the discussion progressed, we realised that some of the awkward tasks that the 500 programmer had to do could also be included in the converter. A few beers later (no doubt) and the concept of a cross-assembler (cross-compiler) came about.

Essentially, we would use symbolic names for commands and for the “cells” which were used for storing data or commands, use the names like a real computer language and then perform some minor bits of arithmetic and bit setting in order to set the computer registers. So the steps from initial coding to getting a punched application on paper tape

## The Core Memory Project

were reduced, plus the luxury of a teletype instead of the full keyboard (12 column wide) tape punches for the NCR 500.

John did nearly all the 4100 code (from what I remember) with me being an avid designer and tester.

Because the NCR 500 had 400 words of 48-bit memory (I wonder why it was called the "500") and the NCR-Elliott had an enormous expanse of 16k (24 bit) we conjured up the cross compiler as being completely in memory.

Like almost all programs there were always fixes needed. We would punch a tape of the corrections and that would be fed in after the original. After a few test cycles we had a load of small bits of tape, that really needed to be inserted in order to make sure that "fixes to fixes" were in the right order! We also had re-usable subroutines. Sterling conversion for example was commonly used to convert British Currency (Pounds, Shillings and Pence) to pence for calculations and storage, then reconvert back for printing. So our STCON subroutine was a pre-punched tape.

Another pub night and we specified "Source Output" where the current master would be punched on high speed tape punches on the 4100. John wrote it that night and it worked the next day first time. So we now had a "clean version" to work from, or archive the final program.

Conveniently the "Banks" developers had now consolidated in St. Alphage House on London Wall where there was also a bureau where the 4100 had moved. So quite a few programs were written and delivered to customers using this convenient technique. We considered the idea of a 500 Simulator to make use of the "spare time" on the 4100 for debugging, but by then the newer units like 399 and Century were coming along. I have read that later that the 399 also had a cross-assembler that could run on the Century.

John and I were awarded a small fee for our idea and the package that was built and used by many programmers in London.

Part of the key to developing a program is coming up with a name. "500" is difficult to build into an acronym in English, but thanks to the Romans, we were able to use "D" in the same to create MIDAS. Method of Interpreting 500 Assembler Symbols.