A NEW DIAGNOSTIC ROUTINE BLOCK

by

J.M. Bennett*, J.C. Butcher*, M. Chapple**
School of Physics, University of Sydney, Sydney. N.S.W.

BACKGROUND

The general policy which is followed in checking out programmes on the SILLIAC is that users are encouraged by the availability of a first-class set of checking routines to spend as little machine time as possible on any single code check. Emphasis is added to this policy by the fact that the monitor tube displaying the contents of any one of the C.R.T. stores is useful only as an indicator of the extent of a "loop". (The orders being obeyed frequently will brighten up in the manner often peculiar to machines with cathode ray tube stores.) As a result, apart from a neon display of the current instruction number and the current instruction pair, the programmer has nothing to peep at, and so peeping is an even more fruitless pursuit here than it is with most computers.

The success of this policy may be gauged from the distribution of code checking times during the month of March, 1957, as shown in Table I. During this period a total of 25 different problems were in the development stage and it will be seen that the mean period taken up for a single code check is about four minutes. Most of the machine time used for development has been taken during half-hour periods during which code checking has priority over production: until very recently there have been two of these periods every day.

Most of the programmes used for code checking came to us as part of the University of Illinois library; and the speed with which our organisation has become effective is due largely to the generosity of the ILLIAC group, for which we are duly grateful.

ILLIAC CODE-CHECKING ROUTINES

The code-checking programmes available are two types - post-mortem routines and diagnostic routines. As the name implies, the first group is used when the programme has "died", i.e. come to a stop on an illegal order combination. The second group is used to diagnose the ills of a patient who, though ailing, is still "alive".

The first post-mortem routine (of which there are a number of variants) is a routine which can be input with a minimum disturbance to the information already in the machine store, which will enable selected sections of the store to be output in sexadecimal form, as fractions, as integers, or as floating point numbers.

The second post-mortem routine carries out a function which is almost identical with that of the SILLIAC's initial orders, the only difference being that an order assembled from tape by this routine is not placed in the store, but instead is compared with the corresponding order already in the store; any difference causes the two versions to be output. In this way it is possible to determine all changes to a programme which have taken place during its execution.

* Adolph Basser Computing Laboratory.
** Snowy Mountains Authority.

The third post-mortem routine is known as an "address search" routine. This is used most frequently when the machine stops on an illegal order. The attempt to obey the illegal order usually results from a transfer of control with an incorrect address, or a transfer of information to an incorrect address in the memory - an address normally occupied by the programme In both cases a knowledge of the orders with the address of the offending illegal order will enable the source of the trouble to be located. This facility is provided by the address search routine.

When it is placed in the store, together with the specification of the pertinent address, it will print out all orders with this address, together with their positions in the store. In case an illegal order is reached only after a certain amount of numerical information has been obeyed as if it represented a sequence of apparently legitimate orders, the programme has been arranged so that the operator can cause it to search for the address previous to that specified, then the one before that, and so on.

The first diagnostic routine is a routine which works on the master slave principle, and records the position of all effective transfers of control, together with a copy of the orders themselves. A pre-assigned block of the store is allocated to the storage of this information, and when this is filled, the routine overwrites the record again, starting from the beginning. Nothing is output, and the information recorded is obtained after the programme has stopped by the use of the post-mortem routine designed to print out sections of the store as order pairs.

This routine is particularly useful in checking out programmes in which blunders are likely to have been made in the logic of the flow diagram rather than in the details of arithmetic.

The second diagnostic routine is called the "Check-Point" Routine. This enables information from any part of the store to be printed out at any stage of the programme, in any one of a variety of forms. Moreover, in the case of a loop of instructions, it provides facilities for suppressing printing during any number of initial executions of the loop, and printing out specified information during any subsequent number of executions of the loop, thereafter suppressing printing altogether.

This routine works in the following manner. When a programme has been input, and just before it has started, the check-point routine is input, together with a specification of all points ("check-points") at which printout is required, and the details of the printout called for. This information is placed in the store after the routine itself, together with the orders which are nominated as check-points. The check-point orders are replaced by transfers of control to orders providing a linking facility, which are stored along with the printout information just mentioned.

When a check-point is reached, the routine is entered and appropriate printout will be effected. When this is completed the stored check-point order will be obeyed, after which the linking arrangements enable control to be returned to the appropriate point in the main programme.

As it stands at the moment, this routine suffers from the major disability that it occupies an undue proportion of the SILLIAC's store. In fact, of the 1024 words available, 57 words are required for the check point routine, and 31+s words are required for the specification of print, when 31 is the number of check-points and s is the number of separate printouts required.

"STOP-AND-PRINT" ROUTINE OF MANCHESTER MARK I MACHINE

In 1951, one of the authors (J.M. Bennett) together with Mrs. C.M.

Berners-Lee (then Miss M.L. Woods) constructed a similar programme for use with the Manchester Mark I machine. This machine had a working store of 512 words and a backing store in the form of a magnetic drum with a total capacity of 16,000 words.

In this case complete specification of printout required and blocking orders was kept on punched paper tape, and the check point routine (which was called "Stop-and-Print") were kept in the backing store. The programme was arranged so that only two words in the working store were associated with the Stop-and-Print Routine, the arrangement being that when the blocking order is encountered, control is transferred to these orders. The section of the store which includes these orders will then be transferred to the backing store and the first section of the Stop-and-Print Routine brought from the backing store into this section of the working store.

At this point the Stop-and-Print Routine takes control; paper tape is taken in via the input reader, specifying for each item of printout the position of the information to be printed, and the form of printout required. This information is acted upon item by item until printout at one check-point is complete; the following check-point is taken in from tape and recognized by an appropriate prefacing symbol.

A blocking order is placed at this new check-point, the order which it displaces being stored along with the Stop-and-Print Routine. The original contents of the section of the store being used by the Stop-and-Print Routine are replaced, and control returned to the main programme. The main programme is then obeyed at full speed until the next blocking order is encountered, at which point the process is repeated.

Although this programme requires the permanent use of only a very small section of the working store, its total size (about 160 words) is quite large. Obviously, in the absence of any form of backing store, a different technique is required with SILLIAC.

A NEW DIAGNOSTIC ROUTINE FOR SILLIAC

The construction of a Check-Point Routine for SILLIAC, which uses a minimum amount of storage space, is now complete. The saving of storage space in comparison with the Check-Point Routine made available to us by the University of Illinois is effected in two ways.

First, the specification of check-points and printout information is no longer stored inside the computer. As in the Stop-and-Print Routine just described, this information is punched on a specification tape and taken into the machine as required.

Secondly, printout is restricted to the simplest possible form, with a bare minimum of ancillary lay-out characters. The form which is simplest for SILLIAC is sexadecimal printout, and this is used throughout. However, this printout in sexadecimal form is regarded as being an intermediate printout only. The specification tape includes, as with the Stop-and-Print Routine, full details of the form in which printout is required, together with various punctuation and lay-out characters. This information is ignored when the specification tape is first passed through the machine, the only information to be acted upon being the position of check-points, and the registers the contents of which are required.

The intermediate printout can be used as it stands: it gives sexadecimal printout in a column lay-out. It can also be inserted into the machine at a later stage (at which stage the whole of the working store will be available) together with a special section of the Check-Point Routine. This section of the routine will take in the specification tape again, this time taking cognizance also of the specification of the form of printout, and punctuation and lay-out characters. The printout from this second section of the routine may be in any one of the following forms:

Ten character sexadecimal word.
Order pair with decimal addresses.
Left hand address (decimal)
Right hand address (decimal).
Signed 12 figure decimal fraction, unrounded.
Signed 12 figure decimal integer, with zero suppression.
Signed 6 figure decimal fraction, unrounded.
Signed 6 figure decimal integer, with zero suppression.
Two half register decimal fractions, unrounded, separated by a comma.
Two half register decimal integers, with zero suppression, separated by a comma.
Floating decimal number, the fractional part to 10 decimal places, unrounded, the two parts separated by a comma.

The first section of this Check-Point Routine requires 26 words only, and this is approximately the storage space required by the machine's initial orders.

In practice, the routine has one disadvantage, in that, if it is required for checking a routine which calls for intermediate input, steps must be taken to ensure that characters being input, and characters designed for the Check-Point Routine, reach the machine in the correct order.

This may be arranged for by requiring that the SILLIAC should stop on entering a routine calling for a normal input. At such a point, the Check-Point specification tape would be replaced by the data tape under test. Of course, the reverse process is required on leaving the input routine.

In view of the fact that the prime motive for constructing the routine was the saving of space, this disadvantage which applies only in a limited number of cases, is not viewed seriously.

TABLE 1

DURATION OF CODE-CHECK RUNS DURING
MARCH, 1957

(Twenty-five users were code-checking during this period.)

Duration of Code Check	Number of Runs
0-1 minute	11
1-2 "	62
2-3 "	91
3-4 "	77
4-5 "	62
5-6 "	33
6-7 "	26
7-8 "	14
8-9 "	9
9-10 "	8
10 "	9*

* Most of these longer runs concern the testing of programmes with representative sets of parameters.