ON STARTING ROUTINES FOR THE C.S.I.R.O. MARK I COMPUTER*

By G. W. HILL† and T. PEARCEY†

Introduction

High speed computers have a normal mode of operation whereby operations are specified by " commands " from stored programmes. This mode cannot be used for starting a cleared computer since there are no stored commands to perform the organized process of assembling commands from the input medium. Additional equipment is therefore required in order to provide another mode of operation for inserting the first, or primary, routine.

Primary Routine

The C.S.I.R.O. Mark I computer can insert a primary routine from stepping switches, as is described in previous papers (Pearcey and Hill 1953*a*, 1953*b*) from which the notation in this article is adopted. Other non-normal modes of operation are included for purposes of testing the computer. One of these modes is used to insert the primary routine, listed in Table 1, from punched tape into the store. The special equipment required for the previous primary routine is therefore redundant.

For testing purposes, commands can be taken from the switchboard of the console, instead of from the store, by special modes of operation controlled from the switchboard. A switch denoted as " S & N_1 to INT " enables performance of the logical sum of sequence register contents (increasing serially) and the command on the switch-set register N_1. If this command is " $(I) \rightarrow M$ " then this mode of operation reads rows of punched tape from the tape reader into serial locations in the store. In this manner the primary routine listed in Table 1 is inserted into the cleared computer.

* Manuscript received May 18, 1955.

† Division of Radiophysics, C.S.I.R.O., University Grounds, Sydney.

Since the input equipment of the C.S.I.R.O. Mark I computer reads 10-hole tape rows into digit positions p_1–p_{10} and the X and Y holes into digit positions p_{19} and p_{20}, respectively, the primary routine must not require digits in the positions p_{11}–p_{18}. This adds a fourth requirement to the three requirements of any primary routine. Thus the primary routine must:

(1) assemble 20-digit commands from two 10-digit tape rows according to the convention that the address-digits row (if any) is followed by an X-punched operation code row;

(2) insert assembled commands in serial locations of the store;

(3) cause Y-punched "control designations" to shift control out of the primary routine for performance of special operations;

(4) be composed of digits in the p_1–p_{10} and p_{19}, p_{20} positions only.

The primary routine set out in Table 1 satisfies all these requirements.

TABLE 1

PRIMARY ROUTINE

Location	Command		Operation
0	$(0) \overset{c}{\rightarrow} S$*	[4]	Command itself contains 3 digits, shifts control to 4
1	$(C) \overset{+}{\rightarrow} K$		Address in C is added to the following command
2	$c(A) \rightarrow 0$		Transfers command in A to store location specified by (C)
3	$p_{11} \overset{+}{\rightarrow} C$		Adds unity to the address in C for serial storage
4	$(Z) \rightarrow B$		Sets $(B)=0$; indicating "non-X" row for command 15
5	$(D_0) \rightarrow H_l$		Sets 10-digit input into H
6	$(D_0) \overset{+}{\rightarrow} D_0$		D_0 contains p_{20} only if X-hole just recently read
7	$s(D_0) \overset{c}{\rightarrow} S$		Tests whether recently read row is X-punched or not
8	$(0) \overset{c}{\rightarrow} S$	[12]	Jumps three commands to 12, to add non-X punched row to upper half
9	$p_{11} \rightarrow B$		Sets $(B)=p_{11}$; indicating "X-row" for command 15
10	$(H_l) \overset{+}{\rightarrow} A$		Adds operation code to p_1–p_{10} positions to assemble command
11	$p_1 \overset{c}{\rightarrow} S$		Jump next command
12	$(H_u) \overset{+}{\rightarrow} A$		Adds address code to p_{11}–p_{20} positions to assemble command
13	$(I) \rightarrow D_0$		Reads tape row into D_0 with p_{19} for X, p_{20} for Y
14	$s(D_0) \overset{c}{\rightarrow} S$		If Y-punched row, jump to command 16
15	$(B) \rightarrow S$	[0, 1]	Shift to 1 or 0 if previous row X-punched or not (resp.)
16	$[(D_0) \rightarrow H_l]$		These four commands are overwritten by subsequent input after being used to set $16p_{11}$ in C by shifting $16p_1$ read from tape through H into upper half of C and then clearing D_0 and S
17	$[(H_u) \rightarrow C]$		
18	$[(D_0) \overset{-}{\rightarrow} D_0]$		
19	$[(Z) \rightarrow S]$	[0]	
	Blank tape rows		Restore normal mode of operation by hand (switch "S & N_1 to INT" off)
	$16Y$		Y-detection by command 14 shifts to special operation of setting the starting address in C

* Command "$(0) \overset{c}{\rightarrow} S$" on the tape is immediately preceded by an X punch.

The first requirement is satisfied since the 10-digit row read from tape by command 13 is added as an upper half address by commands 8, 12 if commands 6, 7 indicate the row is not X-punched, or is added as a lower half operation code by commands 9, 10, 11 if commands 6, 7 indicate the row is X-punched. Serial storage of assembled commands by commands 1, 2, 3, whenever unity in B indicates previous row as X-punched, satisfies the second requirement. Command 14 detects Y-punched rows and shifts control out of the primary routine to command 16, thus fulfilling the third requirement.

To satisfy the fourth requirement, this primary routine uses commands with zero addresses and holds necessary addresses in registers. Sequence shifts, which normally imply addresses, are replaced by the " sequence jump " command " $(0) \xrightarrow{c} S$ ", which jumps three commands for the three digits in the command itself. Register B is used to hold addresses 1 or 0 corresponding to X-hole or blank on the row previously read. The " storage address " in register C is set initially by commands 16, 17, 18, 19 as soon as the routine reads the $16Y$ row and this address is combined with command 2 by use of the " $+K$ " facility to store subsequently assembled commands in serial locations from 16 onwards.

Insertion of the primary routine requires a sequence of switching operations on the console switchboard, which is similar to that required previously for the stepping-switch primary routine.

(1) Clear store. Clear all registers. Switch on reader motor and " tape read " switches.

(2) Set " $(I) \rightarrow T$ " on N_1. Set " S & N_1 to INT ".

(3) Start computer. Computer will stop after initial " X " is read in.

(4) Set " $(I) \rightarrow M$ " on N_1. Clear sequence register and restart.

(5) Just before " $16Y$ " is read, switch " S & N_1 to INT " off, without stopping the machine.

(6) Further commands punched in the normal convention will be stored serially from location 16 onwards.

Control Routine

In most cases, as soon as the primary routine is inserted, it is used to insert a " control routine " from tape. This control routine provides special operations useful in assembling elaborate programmes by suspending command assembly and serial storage while either

(1) the command just assembled is performed instead of being stored,

(2) the " storage address ", n_1, is replaced by n,

(3) the current storage address, n_r, is placed in location n of a parameter block, or

(4) the parameter in location n of the parameter block is added to the next command assembled,

where n is the address number recently read from tape.

The control routine listed in Table 2 is inserted in store locations 16–28 following the primary routine. When a Y-punched " control designation " is detected by command 14, control is shifted to command 16 of the control routine.

The number, x, punched in the " control designation " row is used by commands 17, 18 to shift control to command $19+x$.

For $x \leqslant 3$, constants are added to the address n, previously assembled in A. to form a command, which is set in location 24 and performed. Thus nT, coded as $n;0Y$, sets np_{11} in C so that serial storage will continue from address n. When the storage address is m, the control designation nS, coded as $n;2Y$, sets m in store location $28+n$, i.e. the nth parameter of the block. Subsequently, the control designation nC, coded as $n;1Y$, will reset the transfer address to that held as the nth parameter ; e.g. m in location $28+n$. The designation nA, coded as $n;3Y$, sets the nth parameter, held in location $28+n$, into A so that the next command assembled will be added to this parameter and the sum stored.

Table 2

CONTROL ROUTINE

Location	Command	Operation	
16	$(Z) \rightarrow B$	Clear B to indicate " previous column not X-punched "	
17	$(D_0) \rightarrow H_l$	The " control-designation " number in the Y-punched row is used to select the control operation required	
18	$(H_u) \xrightarrow{+} S$		
19	$(28) \xrightarrow{+} A$	$n\ T$ (transfer), command 24 becomes " $n \xrightarrow{+} C$ "	$n;0Y$*
20	$(27) \xrightarrow{+} A$	$n\ C$ (continue), $(A)+(27)+(26)+(19) \equiv$ " $(28+n) \rightarrow C$ "	$n;1Y$
21	$(26) \xrightarrow{+} A$	$n\ S$ (store), $(A)+(26)+(19) \equiv$ " $(C) \rightarrow 28+n$ "	$n;2Y$
22	$(19) \xrightarrow{+} A$	$n\ A$ (add), $(A)+(19) \equiv$ " $(28+n) \xrightarrow{+} A$ "	$n;3Y$
23	$c(A) \rightarrow 24$	D (do), command in A set in 24 and performed	$;4Y$
24	$[p_{20} \rightarrow T]$	U (unchanged), re-performs previous control operation	$;5Y$
25	$13 \rightarrow S$	N (null), has no effect during insertion	$;6Y$
26	$\langle 0,\ 0;\ 13,\ 27 \rangle$	" $(C) \rightarrow M$ " less " $(M) \xrightarrow{+} A$ "	Constants equal to differences between coded commands
27	$\langle 31,\ 31;\ 18,\ 14 \rangle$	" $(M) \rightarrow C$ " less " $(C) \rightarrow M$ "	
28	$\langle 31,\ 4;\ 26,\ 0 \rangle$	" $(K) \rightarrow C$ " less " $(28) \rightarrow C$ "	

* Punch codes for control designations.

A command followed by D, punched as $4Y$, is set by command 23 into 24 and performed. The last control operation performed may be repeated by using the designation U, punched as $5Y$. The null operation, N, punched as $6Y$, has no effect during insertion of commands, but incidentally is used by the routine for printing punched tape as symbolic commands to provide spaces between printed routines.

These control operations are similar to those used in association with the stepping switch primary routine. However, the number x of the control designation is now to be punched as a binary number rather than as x holes in a row. Also the operation of replacing the storage command, R, has been changed to the operation, C (continue), which restores the storage address to that held in location n of the parameter block. The main use of operation R was for replacing the storage command in order to insert programmes in the backing store instead of the high-speed command store. This effect can be obtained by the sequence of operations :

$$c(A) \rightarrow 2, \text{ D}$$
$$c(A) \rightarrow M', \ U.$$

The primary and control routines listed in Tables 1 and 2 provide all the facilities furnished by the previous starting routines while the store space occupied is reduced from 32 to 29 commands because the " $+K$ " function is used. Since these routines can be inserted without using equipment such as stepping switches, not otherwise required for testing purposes, considerable economy in equipment is possible, and the starting procedure is simplified.

References

PEARCEY, T., and HILL, G. W. (1953*a*).—Programme design for the C.S.I.R.O. Mark I computer. I. Computer conventions. *Aust. J. Phys.* **6**: 316–34.

PEARCEY, T., and HILL, G. W. (1953*b*).—Programme design for the C.S.I.R.O. Mark I computer. II. Programme techniques. *Aust. J. Phys.* **6**: 335–56.