SILLIAC CODE X21

TITLE — Program Interruption Sexadecimal Dump, with Sum Check.

TYPE — Closed routine (entered with link in A).

NUMBER OF WORDS — 42.

TEMPORARY STORAGE — Internal.

DURATION — 230 milliseconds for each location dumped. (All locations are dumped except those containing zero).

PURPOSE — Programs which require more than a half hour continuous computing time must be arranged so that they can be taken off the computer at any time, to be read in and re-started at a later time. Such long programs should also have some programmed check on their operation; sometimes an overall arithmetic check is possible, in other cases a store sum check suffices.

A program can be taken off the computer, to be re-started later, if it punches out the contents of all storage locations that contain information which varies during the running of the program. This operation is called "dumping the program". The program can then be re-started at a later time by reading in the "dump" which overwrites part of the program and restores the contents of the store to the form it had when the program was dumped.

CODE X21 provides a very simple means of dumping a program. It dumps the entire store, thereby eliminating the need for the programmer to plan which part of the program should be dumped, and also eliminating the need to read in the original program before reading in the dump when the program is to be re-started. However, this facility carries

with it the disadvantage that the time taken to dump the program is greater than it would otherwise be. CODE X21 dumps only non-zero locations, hence the time taken is proportional to the size of the program. A program filling the entire store takes about 4 minutes to be dumped. Consequently X21 should be used only with programs which use a small part of the store.

Programs of long duration should be dumped every ten to fifteen minutes, preferably immediately after an arithmetic or sum check has been successfully performed. If such a check fails the program should be brought to a stop without a dump being made. The program can then be restored to its state before the error occurred by reading in the previous dump.

In addition to the automatic dump that occurs at ten or fifteen minute intervals there should be provision for the operator to force a dump to occur at any time, thus enabling her to allow the program to continue right up to the time when the computer is required for another computation.

DESCRIPTION

X21 is entered with the link address in the right hand side of the accumulator, i.e. with the orders

```
p     --    --
      F5    pL
p+1   26    xF
      --    --
```

where xF is the location of the first word of X21.

When entered it punches out the contents of every store location except those which contain zero, then returns control to the right hand side of word p+1.

The tape punched out by X21 starts with a bootstrap sum-checking input routine. This enables the "dump" to be read back into SILLIAC, at any later time. When this is done the contents of the store will be restored exactly the same as they were when the dump was made. As zero locations are not dumped, the store must be cleared before the dump tape is bootstrapped in.

The bootstrap tape ends with a "start" order which transfers control to X21, which then links back to the program at word p+1, enabling the program to proceed from exactly the point at which the store was dumped. Hence, if a program does not use the magnetic tape store, no special provision need to be made to enable it to be dumped by X21, except the inclusion of X21 in the program.

If a program which uses the magnetic tape is to be dumped some provision must be made for the fact that the tape may not be in the same place when the program is restored. The tape will, in fact, be in the rewound condition when the program is replaced in the computer. If the program is dumped only at such a place in the program that the next magnetic tape instruction is a search in the forward direction then it will not matter that the tape is rewound.

METHOD OF USE

Following is an example of one way in which X21 may be used. This coding should follow immediately after the arithmetic check or sum check which verifies that the program has proceeded satisfactorily up to this point.

```
q       -- --
        F5 cF       (c) = Counter

q+1     40 cF
        32 q+2L

q+2     24 q+5L
        F5 q+2L

q+3     26 xF       (x) = First word of X21
        24 q+4L

q+4     L1 aF       (a) = Count constant.
        40 cF

q+5     -- --
```

Suppose that the arithmetic or sum check is performed every 15 seconds. Then, as we want the program to be dumped every 15 minutes, it should be dumped after every 60 checks. Location aF should then contain the integer 60, and location cF initially contain the integer -60.

The program is run with the black switch set down. After every check, location cF is counted up by unity. As long as it remains negative control passes on to word q+5, by-passing the entry into X21. When cF becomes zero, control passes to X21, the program is dumped, -60 is again placed in location cF, and the program proceeds from word q+5 onwards.

When the operator wishes to remove the program from the computer she centres the black switch. If this happens just before a dump is due to occur, it will be punched in the usual way and the program will stop on the 24 order in word q+3. If a dump is not due the program will stop on the 24 order in word q+2. The dump will then be punched if the white switch is raised and lowered, ILLIAC again stopping on the 24 order in word q+3.

If the operator should force a dump in this way, then decide that the program may proceed after all, lowering the black switch again will allow it to proceed. The counting of every 60th check proceeds anew from this point.

ADDITIONAL NOTES

X21 dumps only non-zero locations in the store. Therefore care must be taken to clear the store before a dump is bootstrapped in.

The output tape bears 16 Figure Shifts, a "miniature bootstrap", a sum-checking input routine and its start order, followed by the dump. Each location is punched as a CR/LF, a three character sexadecimal address, and a ten character sexadecimal word. The last word on the tape is the check-sum. Failure of the sum-check will result in a -F stop. If this stop is bypassed with the white switch the program will proceed as though the sum-check has been successful.

Coded by: M.J. Campbell

June 1960.

LOCATION	ORDER	NOTES
0	42 41L	Punch 16 figure-shifts.
	92 765F	
1	50 22L	Output the bootstrap.
	F5 1L	
2	26 10L	
	39 F	
3	50 23L	
	F5 3L	
4	26 10L	Output the input routine.
	1S 38F	
5	50 25L	
	06 28F	
6	80 52F	Output transfer of control
	41 31L	to input routine. Clear sum.
7	50 24L	
	F5 7L	
8	26 10L	Output the remainder of the
	L5 31L	store.
9	82 40F	Output the sum.
	22 41L	
10	NO 30L	From 2,4,8.Output Routine.
	42 21L	
11	3L 30L	From 21.
	L3 F	
12	36 19L	Is the location empty?
	92 129F	No. Punch CR/LF.
13	L5 30L	
	32 17L	Is the address to be punched?
14	02 28F	Yes.
	82 12F	Punch 3-digit address.
15	-5 F	
	3L 30L	
16	L4 F	
	L4 31L	
17	40 31L	Form the sum.
	3L 30L	From 13.
18	L5 F	
	82 40F	Punch contents of location.
19	F5 30L	From 12.
	40 30L	Increment location counter and
20	02 20F	test for end-point.
	L0 30L	
21	36 11L	
	22 ()F	By 10.

LOCATION	ORDER		NOTES
22	00 29L	By 1	Parameters for output of bootstrap.
	80 26L		
23	80 42L	By 3	Parameters for output of input routine.
	00 32L		
24	80 1053L	By 7	Parameters for dumping store.
	00 43L		
25	41 29L	By 5	Transfer of control to input routine. (Placed in location 1)
	26 32L		
26	80 40F		Bootstrap.
	40 1F		
27	80 40F		
	40 2F		
28	41 31L		
	26 F		
29	80 12F		
	42 F		
30	00 F	By 10,11,13,	Location counter and temp.
	00 F	15, 17, 19,	storage.
		19, 20, 34,	
		36.	
31	00 F	By 6, 8, 16,	Check-sum.
	00 F	17,20,36,37,	
		39,39, 40.	
32	41 F		Input Routine. Clear
	41 1F		locations used by bootstrap.
33	41 2F		
	81 12F	From 38.	Read address.
34	40 30L		
	42 35L		
35	80 40F		Read.
	40 ()F	By 34.	Store.
36	L4 31L		
	L4 30L		
37	40 31L		Form sum.
	L3 29L		
38	32 33L		Input until last location is filled.
	60 40F		Read sum.
39	L0 31L		
	40 31L		
40	L3 31L		Test for sum failure.
	32 41L		
41	-F F		
	22 ()F	By 0. From 9, 40.	