bcc	title	prefix/class-number.revision
	DCODT REFERENCE MANUAL	DCODT/M- 20

checked	authors	approval date	revision date
checked	Butler Lampson	5/19/70	
approved		classification: Manual	
		distribution: Company Private	pages: 7

ABSTRACT and CONTENTS

This is a reference manual for DCODT, a microcoded octal DDT intended to be used for system maintenance and checkout of the DCC.

DCODT is a two-board microcoded octal DDT designed to run on the DCC and intended to be used for DCC maintenance and for system checkout of DCC's. The final version will exist in the DCC maintenance hardware and will make use of this hardware's teletype interface and breakpoint facilities. A preliminary version uses the standard microprocessor teletype interface and has no breakpoint capability.

DCODT offers facilities for examining and modifying the state of core in a variety of ways. It also permits the entire microprocessor state to be set up from core (scratchpad, holding registers, M, Q, Z, OS, A) and the microprocessor to be started at any location. Whenever it is entered (by breakpoint or reset) it saves the state in core (except for scratchpad location 8 and the O register) in such a way that it will be restored when the micro-program is restarted. The user will be able to continue as if no breakpoint occurred (except for timing problems), possibly after modifying the state.

A. Examining Memory

1) Typing

n

to ODT will cause the contents of location n (an octal number) to be typed out in octal. Leading zeros are suppressed in this typeout.

2) /

alone will type the contents of the location addressed by the last thing which was typed.

3) ,

will type

n/c

where n-1 is the last location whose contents was typed, and c is the contents of location n

4) ↑

will type

n/c

where n+1 is the last location whose contents was typed, and c is the contents of location n.

It is also possible to type things in formats other than octal.

5) n\

will type a space, but ODT will thereafter behave as though the contents of location n had been typed.

6) n:

will type the contents of location n in the format

b;c

where b is the first 8 bits and c the last 8 bits.

7) n]

will type the contents of location n as an RPU instruction (see RPU/S-33). There are two formats. For an ordinary instruction what is typed is:

o^r±a

where o is the opcode (bits 6-9), r the register (bits Ø-2), a the address (bits 10-15), the + or - indicates the sign (bit 3) and ^ indicates the addressing mode or tag (bits 4-5) as follows:

.	direct	tag = 3
=	immediate	tag = 2
(	scratchpad	tag = 1
*	indirect	tag = Ø

Thus 16*2+14

would be a branch to the 14th location from this one.

An extended operation (bits 6-9=Ø) is printed as

oijX

where o is the extended opcode (bits 11-15), i is the i register (bits Ø-2) and j is the j register (bits 3-5).

Once a location has been examined with \ : or], subsequent uses of , or ↑ will preserve the typeout mode until the next occurrence of /, \, : or] or a reentry to ODT via break or reset.

B. Changing Memory

Typing

1) n↵

(where ↵ stands for carriage return) stores n into the last location whose contents were typed. This works at any time. n may be an octal number, or it may be in the form typed out by : or]. Exception: the last bit of the number before a; in the two-byte format is zeroed. Sorry.

2) n, or n↑

stores n into the last location whose contents were typed before performing the , or ↑ function.

Spaces may be typed at any time for readability. They are echoed but otherwise ignored.

C. Referencing the Stored State

ODT stores the microprocessor state in high core at addresses listed in Appendix I. Most of these addresses can be specified symbolically, so that the user does not have to remember or type the actual numeric address.

1) M, Q, Z are the addresses of the stored M, Q and Z registers.
2) L is the address of the stored OS (Link) register
3) nR is the address of the stored nth holding register ($0 \leq n \leq 3$)
4) nS is the address of the stored nth scratchpad ($0 \leq n \leq 77$). Of course, n is octal.

The stored value of the A flag does not have a symbolic name. It is at Z+1.

D. Running the Microprogram

Whenever ODT is entered it stores the microprocessor state in core at the locations specified in Appendix I. It then goes through the autoprint sequence (see E below) and listens for input. This can happen as a result of a reset of the microprocessor or of a break. In the latter case the address of the instruction which was about to be executed (the one in the break address switches) is preserved in a hardware register B. The address of the next instruction (which may not be B+1 if a DGOTO is pending) is preserved in another hardware register N. The B and N registers cannot be examined or altered by ODT.

The command

    nG

loads the microprocessor state from core and sends control to location n.

The command

    C

loads the microprocessor state from core and does

    DGOTO B
    DGOTO N

If the state which was saved at the last break has not been altered, this restores the entire microprocessor to the state it was in at the time of the break. At least one instruction will be executed before the next break.

E. Autoprint

A core location with the symbolic name @ contains the autoprint address (AA). Whenever ODT is entered, after storing the state it fetches AA. If AA is Ø nothing is done. Otherwise the contents of location AA is printed in octal. To set up autoprint to display Z after each break, for example, just type @\ZØ.

F. Reading Programs into ODT

If a sequence of core locations is typed out with , or ↑ and the paper tape punch turned on, the resulting tape will reconstruct those core locations if it is read back into ODT. Thus

```
,
100/  123,
101/  456,
102/  701,
```

on the tape can be read in, preceded by typing

77\

and the desired effect will be obtained.

Important Note:

When typing in b;c exactly three digits of c must be typed. When typing in 0⌂r±a exactly two digits of a must be typed. Because of this the current version of ODT will not read back a core image dumped in instruction formats since the address field may not be two digits long. It is planned to fix this later.

APPENDIX I: Core Locations for the Saved State

The addresses are given for a 4k memory. Add 10000 for an 8k memory, since the state is always stored at the top of core.

SKØ - SK77	7700-7777
RØ - R3	7670-7673
M	7710
Q	7674
Z	7675
A	7676
OS	7677
@ (autoprint)	7667
clobbered by ODT	7666

No other core locations are altered by ODT except at the explicit request of the user.

APPENDIX II: Summary of ODT Commands

Here $ stands for the last location whose contents was typed, and ($) for the contents of location $. An octal number is represented by n, and the contents of location n by (n).

n/	type (n) in octal
/	type (($)) in octal (also for :,],\)
n:	type (n) as two bytes
n]	type (n) as RPU instruction
n\	$←n
,	type ($+1)
↑	type ($-1)
n⌀	store n in location $
n,	store n in location $, then type ($+1)
n↑	store n in location $, then type ($-1)
b;c	number with two 8-bit bytes b and c
o.r±a	direct addressing RPU instruction with opcode o, register r, address a, sign Ø(+) or 1(-1)
o*r±a	indirect addressing RPU instruction
o=r±a	immediate addressing RPU instruction
o(r±a	scratchpad addressing RPU instruction
oijX	extended RPU instruction with extended op o.
M,Q,Z,L,nR,nS	core addresses for stored M,Q,Z,OS(link), holding and scratchpad registers
@	core address for autoprint address
nG	restore state and go to location n
C	restore state and continue after breakpoint
space	echoed and ignored
any other character	ignored