TX-0 COMPUTER
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE 39, MASSACHUSETTS

March 24, 1961

To TX-0 Users:

Enclosed is a memo describing a new utility routine, designed to replace UT-3. The changes taking place in TX-0's order code will obsolete UT-3, but it was felt that a utility routine occupying little more memory than UT-3 was desirable.

At present, the relocatable version of FLIT JR. is still in preparation. Therefore, two binary versions of the routine, occupying different memory locations, were prepared.

The first, FLIT JR. occupies registers 5533 to 10240. It is entered at register 6000. FLIT JR. HIGH, substantially the same routine, occupies registers 15233 to 17777. Its entry point is register 15400.

It is hoped that users will find these routines a satisfactory replacement for UT-3.

JBD:EAB

Jack Dennis eab

J. B. Dennis

TX-0 COMPUTER
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE 39, MASSACHUSETTS

M-5001-32

March 24, 1961

UTILITY PROGRAM FLIT JR.

FLIT JR. is a utility program for TX-0 containing a subset of the features of FLIT and occupying about half as many or 2500_8 registers. It should prove convenient for users whose programs do not leave sufficient space for FLIT. FLIT JR. will only work with octal numbers and will not accept symbol definitions beyond its permanent vocabulary, which is the same as for FLIT and MACRO. A list of all features of FLIT JR. is attached. FLIT JR. is relocatable, allowing the user to place it in memory starting from any desired location. To store FLIT JR. starting at register y , place the instruction trn y in the TBR, place FLIT JR. in the PETR and press the read in button. Subsequently, pressing test will return control to FLIT JR.

One useful additional feature has been provided. Typing

proceed n

where n is an octal number will cause the breakpoint to be ignored the first n times computer control reaches it.

The input instruction in FLIT JR. will punch the sequence

[tape feed]

tra 17744

[tape feed]

A tape punched in this manner, and placed in the PETR will transfer control to an input routine already in memory, when the read in button is pressed.

R. A. Wagner

R. A. Wagner

Jack Dennis

J. B. Dennis

PSEUDOINSTRUCTIONS AND CHARACTER MEANINGS

A. Pseudoinstructions:

instructions	type as instructions
constants	type as constants
print a,b	print registers a to b horizontally
word w	search for w
word w,a,b	search for w from a to b
word w,a,b,m	search for w from a to b masked by m
address l	search for address l
address l,a,b	search for address l from a to b
address l,a,b,m	search for address l from a to b masked by m
feed	feed three inches blank tape
input	punch input routine
punch a,b	punch memory from a to b
start l	punch start block
start add l	punch automatic start block
begin l	start program at l
begin l,ac	start program at l with ACC = ac
begin l,ac,lr	start at l with C(ACC) = ac and C(LR) = lr
break bp	stop when breakpoint reached
proceed	proceed from last breakpoint
proceed n	ignore breakpoint the first n times computer control reaches it
break	erase breakpoint

B. Characters:

→|	make modification if register is open
↵	close register and make modification
←	close register, open next
=	equals as a constant
I	equals as an instruction
/	register referred to contains the instruction
(	register referred to contains the constant
▨	plus
+	plus
-	minus
F	address of last free register
W	last word typed
L	last register opened
|	open register specified; or allow comment
P	punch new contents of register last examined
delete	delete all typing since last tab or carriage return
,	separates pseudoinstruction arguments;