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

M-5001-29

THE FUTURE TX-0 INSTRUCTION CODE

September 23, 1960

I. ABBREVIATIONS

AC	:	ACcumulator
LR	:	Live Register
PC	:	Program Counter
MBR	:	Memory Buffer Register
PETR	:	PhotoElectric Tape Reader
TAC	:	Toggle Switch Accumulator
TBR	:	Toggle Switch Buffer Register
LP1	:	Light Pen FF
LP2	:	Light Gun FF

C(AC): "Contents of AC"

$\longrightarrow$: "Replaces"

$\overline{X}$: Complement of X:

X	$\overline{X}$
0	1
1	0

$\cap$: Intersect; and; logical product.

$\cup$: Union; inclusive or; logical sum.

$\wedge$: Partial add; inequivalence; exclusive or.

mod n: Modulo n ; $y = x \bmod n$ means $x = kn + y$ for some integer k, $0 \leq k \leq n - 1$.

x	y	$x \cap y$	$x \cup y$	$x \wedge y$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

II. ADDRESSABLE COMMANDS

(A) STORE CLASS

MNEMONIC	OCTAL VALUE	OPERATION	SYMBOLIC DESCRIPTION
STO y	000000 + y	Store	$C(AC) \rightarrow C(y)$
Place the contents of AC in register y. The previous contents of y are destroyed. Contents of AC remain unchanged.			
STX y	020000 + y	Store AC, Indexed	$C(AC) \rightarrow C(z)$ $z = y + C(XR)$ *
Form an effective address, z, by adding * C(XR) to y.			
SXA y	040000 + y	Store Index in Address	$C(XR)_{5-17} \rightarrow C(y)_{5-17}$
Store the digits of the index register in the address portion of register y. The sign of XR is ignored. The contents of XR are unchanged. Bits 0 through 4 of register y are unchanged.			
ADO y	060000 + y **	Add One	$C(y) + 1 \rightarrow C(AC)$ $C(y) + 1 \rightarrow C(y)$ **
Add One to the contents of memory register y and leave the result in the accumulator and register y.			
SLR y	100000 + y	Store Live Register	$C(LR) \rightarrow C(y)$
The contents of LR are placed in register y. The previous contents of y are destroyed. Contents of LR are unchanged.			
SLX y	120000 + y	Store LR, Indexed. (See STX)	$C(LR) \rightarrow C(z)$

* One's complement addition of 14-bit quantities.

** One's complement addition of 18-bit quantities.

[illegible]. INSTRUCTION CHARTS

([illegible]) ADD CLASS

MNEMONIC	OCTAL VALUE	OPERATION	SYMBOLIC REPRESENTATION
ADD y	200000 + y	Add the contents of register y to AC. Contents of y are unchanged.	$C(y) + C(AC) \to C(AC)$ **
ADX y	200000 + y	Add, indexed. (See STX)	$C(z) + C(AC) \to C(AC)$ **
LDX y	240000 + y	Load Index Load the index register from bit 0 and bits 5 through 17 of register y. The contents of y are are unchanged. This instruction places the signed contents of a memory register in the index register provided its magnitude is within the capacity of XR, that is $\lvert C(y) \rvert \leq 2^{13} - 1$.	$C(y)_{5-17} \to C(XR)_{5-17}$ $C(y)_0 \to C(XR)_0$
AUX y	260000 + y	Augment Index The contents of memory register y are added * to XR. The fourteen bit number added consists of bit 0 and bits 5 through 17 of register y.	$C(y)_{0,5-17} + C(XR) \to C(XR)$ *
LLR y	300000 + y	Load Live Register The contents of register y replace the previous contents of LR. Contents of y are unchanged. Previous contents of LR are destroyed.	$C(y) \to C(LR)$
LLX y	320000 + y	Load LR, indexed (See STX)	$C(z) \to C(LR)$

* One's complement addition of 14-bit quantities.

** One's complement addition of 18-bit quantities.

II. [illegible]

(6) [illegible]

MNEMONIC	OCTAL FORM	OPERATION	CONDITIONS FOR TRANSFER $y \rightarrow C(PC)$	[illegible]
[illegible] y	400000 + y	Transfer on Negative AC If the AC bit 0 is a one, take next instruction from register y. Otherwise, take next instruction in sequence.	$C(AC)_0 = 1$	$C(AC)_0 = $ [illegible]
TZE y	420000 + y	Transfer on Zero If the contents of the accumulator are either plus zero or minus zero, the next instruction is taken from register y. If the accumulator contents are not plus or minus zero, the next instruction in sequence will be executed.	$C(AC) = +0$ or $C(AC) = -0$	$C(AC)$ [illegible] and $C(AC)$ [illegible]
TSX y	440000 + y	Transfer and Set Index The next instruction is taken from register y and the address of the register following the TSX instruction is placed in the index register.	Always; $C(PC)+1 \rightarrow C(XR)_{[illegible]}$ $0 \rightarrow C(XR)_0$	Never
TIX y	460000 + y	Transfer and Index If the index register contains plus or minus zero, perform the next instruction in sequence without changing the contents of the index register. If the index register contains a positive number, its contents are reduced by one and the next instruction is taken from register y. If the index register contains a non-zero negative number its contents are increased by one and the next instruction is taken from register y. A zero result will have the same sign as the initial contents of the index register.	$C(XR) \neq +0$ and $C(XR) \neq -0$ If $C(XR)_4 = 1$, $C(Xk)+1 \rightarrow C(XR)$; If $C(XR)_4 = 0$, $-[-C(XR)+1] \rightarrow C(XR)$	$C(XR) = +0$, or $C(XR) = -0$
TRA y	500000 + y	Transfer The next instruction is taken from Register y.	Always	Never
TRX y	520000 + y	Transfer Indexed. (See TSX)	Always $z \rightarrow C(PC)$	Never

III. OPERATE CLASS COMMANDS

(A) MICRO ORDER CHART

Instruction bits

		2	3	9	10	11	12	13	14	15	16	17
Cycle Zero	7		1 AMB									
	8	1 CLA										
				In - Out Stop								
Cycle One — Time Pulses	2			0	X XMB	1	1 COM					
	3									1	1 ANB	1
										1	0 ORB	1
	4			0	1 MBL	X				0	1 LMB	X
	5							1 PAD				
	6			1	0 SHR	0						
				1	1 CYR	0						
	7								1 CRY			
	8									0	X MBX	1

X - Bit may be either zero or one

[illegible]

[illegible]

[illegible]

	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible] of the commands [illegible] [illegible] [illegible] the [illegible] commands [illegible] [illegible]

[illegible] and [illegible] will remain operable while the [illegible] and [illegible] logic is [illegible]

[illegible] See Memo M-5001-18, "[illegible]"

III. OPERATE CLASS COMMANDS

(c) MICRO-ORDERS

MNEMONIC	ACTION	SYMBOLIC DESCRIPTION
CLA	CLear AC	$0 \rightarrow C(AC)$
AMB	transfer AC contents to MBR	$C(AC) \rightarrow C(MBR)$
XMB	transfer XR contents to MBR	C(XR) bits 5-17 → C(MBR) bits 5-17. C(XR) bit 0 → C(MBR) bits 0-4.
MBL	transfer MBR contents to LR	$C(MBR) \rightarrow C(LR)$
LMB	transfer LR contents to MBR. Note: LMB & MLR, if used simultaneously, interchange C(LR) and C(MBR).	$C(XR)_{5-17} \rightarrow C(MBR)_{5-17}$ $C(XR)_{4} \rightarrow C(MBR)_{0-4}$
MBX	transfer MBR contents to XR	$C(MBR)_{5-17} \rightarrow C(XR)_{5-17}$ $E(MBR)_{0} \rightarrow C(XR)_{4}$
CYR	CYcle AC contents Right one binary position. (AC bit 17 goes to AC bit 0)	$C(AC)_{i} \rightarrow C(AC)_{j}$ $i = 0, 1, \ldots, 17$ $j = (i+1) \bmod 18$
SHR	shift AC contents right one binary position (AC bit 0 is unchanged, bit 17 is lost)	$C(AC)_{i} \rightarrow C(AC)_{i}+1,$ $i = 0, 1, 2, \ldots 16$
ANB	ANd (logical product) LR contents into MBR.	$C(LR) \cap C(MBR) \rightarrow C(MBR)$
ORB	OR (logical sum) LR contents into MBR.	$C(LR) \cup C(MBR) \rightarrow C(MBR)$
COM	COMplement AC	$\overline{C(AC)} \rightarrow C(AC)$
PAD	Partial ADd MBR to AC (for each MBR one, complement the corresponding AC bit.)	$C(MBR) \wedge C(AC) \rightarrow (AC)$
CRY	A CarRY digit is a ONE if in the next least significant digit, either AC = 0 and MBR = 1, or AC = 1 and carry digit = 1. The carry digits so determined are partial added to the AC by CRY. PAD and CRY used together give a full one's complement addition of C(MBR) to C(AC).	$CRY\,[C(AC),\ C(MBR)] = C(AC) \wedge C \rightarrow AC.$ $C_{i} = [C(MBR)_{j} \cap \overline{C(AC)_{j}}]$ $\cup [C_{j} \cap C(AC)_{j}]$ $i = 0, 1, \ldots, 17$ $j = (i+1) \bmod 18.$ $CRY\,[C(AC) \wedge C(MBR),\ C(MBR)]$ $= C(AC) + C(MBR)$

III. OPERATE CLASS COMMANDS

(D) IN-OUT GROUP COMMANDS WHICH CAN BE USED WITH MICRO-ORDERS SPECIFIED BY BITS 9-17.

OCTAL CODE	MNEMONIC	ACTION	CYCLE AND TIME PULSE
631000	CLL	CLear Left 9 bits of AC	0.6
632000	CLR	CLear Right 9 bits of AC	0.6
607000	SPF	Set Program Flag contents from MBR	1.6
606000	RPF	Read Program Flag contents into MBR. (inclusive or)	1.2
602000	TBR	transfer TBR contents to MBR. (inclusive or)	1.2
601000	TAC	transfer TAC ones to AC (inclusive or)	1.1
603000	PEN	set AC bit 0 from light PEN FF, and AC bit 1 from light gun FF. (FF's contain one if pen or gun saw displayed point). Then clear both light pen and light gun FF's.	1.1
620000	CPY	CoPY synchronizes transmission of information between in-out equipment and computer	**
621000	R1L	Read ONE Line of tape from PETR into AC bits 0, 3, 6, 9, 12, 15 with CYR before read	IOS
623000	R3L	Read THREE Lines of tape from PETR into AC bits 0, 3, 6, 9, 12, 15, with CYR before each read	IOS
622000	DIS	DISplay a point on scope (AC bits 0-8 specify X coordinate, AC bits 9-17 specify Y coordinate) NOTE: Scope coordinate (0,0) is at center of scope	IOS
626000	P6H	Punch one SIX-bit line of Flexo tape (without seventh hole) from AC bit 2, 5, 8, 11, 14, 17. NOTE: Lines without seventh hole are ignored by PETR	IOS

OCTAL CODE	MNEMONIC	ACTION	TIME PULSE
627000	P7H	same as P6H, but with SEVENTH hole	10S
625000	TYP	read one character from on-line TYPewriter into LR bits 12 to 17.	10S
610000 through 617000	EX0 through EX7	operate user's EXTernal equipment	10S
600000	NOP	perform No in-out group OPeration	
630000	HLT	HaLT the computer (chime sound).	1.8

IV. OPERATE CLASS INSTRUCTIONS

TO BE RECOGNIZED BY MACRO AND FLIT

MNEMONIC	OCTAL VALUE	OPERATION
opr	600000	No operation.
cll	631000	Clear left half of accumulator to zero.
clr	632000	Clear right half of accumulator.
cla	700000	Clear entire accumulator to +0.
clc	700040	Clear and complement: set accumulator to -0.
lro	600200	Clear live register to +0.
xro	600001	Clear index register to +0.
cal	700200	Clear accumulator and live register to +0.
cax	700201	Clear accumulator and index register to +0.
alr	640200	Place accumulator in live register.
alo	640220	ALR, then set AC to +0.
alc	640260	ALR, then set AC to -0.
axr	640001	Place accumulator in index register.
axo	640021	AXR, then set AC to +0.
axc	640061	AXR, then set AC to -0.
xlr	600300	Place index register in live register.
lac	700022	Place live register in accumulator.
lcc	700062	Place complement of live register in accumulator.
xac	700120	Place index register in accumulator.
xcc	700160	Place complement of index register in accumulator.
cyl	640030	Cycle accumulator left one place.
cyr	600600	Cycle accumulator right one place.
shr	600400	Shift accumulator right one place, bit 0 remains unchanged.
lal	700012	Place live register in accumulator cycled left once.
lar	700622	Place live register in accumulator cycled right once.
xal	700110	Place index register in accumulator cycled left once.
all	640230	Place accumulator in live register, then cycle AC left once.
alx	640031	Place accumulator in index register, then cycle AC left once.
ial	740222	Interchange accumulator and live register.
ixl	600303	Interchange index register and live register.
lad	600032	Add live register to accumulator.

MNEMONIC	OCTAL VALUE	OPERATION
xad	600130	Add index register to accumulator.
iad	640232	Interchange and add: accumulator is placed in the live register and the previous contents of the live register are added to it.
anl	640207	Logical and of AC and LR is placed in LR.
ano	740207	ANL, then clear AC.
ana	740027	Logical and of AC and LR is placed in AC.
orl	640205	Logical or of AC and LR is placed in LR.
oro	740205	ORL, then clear AC.
ora	740025	Logical or of AC and LR is placed in AC.
rxa	700322	Place LR in AC, then place XR in LR.
rax	640203	Place LR in XR, then place AC in LR.
lcd	600072	Contents of LR minus those of AC are placed in AC.
xcd	600170	Contents of XR minus those of AC are placed in AC.
com	600040	Complement the accumulator.
lpd	600022	Logical exclusive or of AC and LR is placed in AC. (partial add)
cry	600012	Carry. Result of this operation is same as if contents of LR were added to exclusive or of AC and LR. Partial add and carry result in a full add.
tac	601000	Contents of test accumulator are or'ed to those of AC.
tbr	602000	Contents of test buffer register are or'ed to those of the memory buffer register.
pen	603000	Contents of pen flip-flops 1 and 2 replace contents of AC bits 0 and 1. Pen flip-flops are cleared.
rpf	706020	The program flag register is placed in the accumulator.
cpf	607000	The program flags are cleared.
cpy	620000	Transmits information between the live register and selected input-output unit.
rlc	721000	Read one line paper tape into accumulator bits 0, 3, etc.
r3c	723000	Read three lines of paper tape.
dis	622000	Display point on CRT corresponding to contents of AC.

MNEMONIC	OCTAL VALUE	OPERATION
dso	662020	DIS, then clear accumulator.
prt	624000	Print one on-line flexo character from bits 2, 5. etc. of AC
pnt	624600	PRT, then cycle AC right once to set up another character.
pno	664020	PRT then clear accumulator.
p6h	626000	Punch one line of paper tape 6 holes from bits 2, 5, etc. of AC then cycle right once.
p7h	627600	Same, but punch 7th hole.
p6s	726000	Clear accumulator and punch a line of blank tape.
p6b	766020	Punch a line of blank tape but save the accumulator.
hlt	630000	Stops computer
typ	625000	Read one character from on-line flexowriter into LR bits 12 through 17.
p6o	666020	p6h then clear AC.
p7o	667020	p7h then clear AC.