Web Extra: The Second Life of ENIAC

Hans Neukom

Editor's Note

The following information accompanies the main article, "The Second Life of ENIAC," by Hans Neukom, which appears in the April-June 2006 issue of *IEEE Annals of the History of Computing*. The material here was omitted from the main article because of space constraints.

ENIAC's orders (instructions)

Richard F. Clippinger, head of the Computing Laboratory of the Ballistic Research Laboratory, was responsible for the technical implementation of the ENIAC order set. Table 1 lists all 67 orders implemented in the two initial order sets of 60 orders each.

The table has been edited from the unedited version of the table contained in the Web posting of R.F. Clippinger's paper on the logical coding system of the ENIAC (http://ftp.arl.army.mil/~mike/comphist/48eniac-coding/sec4.html/).

The seven orders that are different in the alternate order set are numbered 73a, 74a, 91a, 92a, 93a, 94a, and 95a. Order numbers 73, 74, 91, 92, 93, 94, and 95 belong to the first set.

List of ENIAC orders of the converter code

In the full set of 94 orders implemented by 1951, the extended order code space (see the section "Upgrading ENIAC" in the main article in the *Annals*) made it possible to define a single set of orders, containing all orders that ENIAC required. The order set could also be simplified using the extended space: the two previous shift instructions were replaced by single shift instructions for each shift direction and count.

Table 2 shows the 94 orders implemented after the new 10-by-10 converter had been added to the ENIAC replacing the original 10-by-6 converter.

The more important changes to the orders, however, concerned the use of accumulators and the specification of order addresses. Order addresses had three decimal digits and covered the 100-to-499 range. The first digit indicated which of the four function tables to use; the other two digits pointed to a line within the respective function table. Because one such line of 12 decimal digits could store six order codes, the address didn't actually point to a specific order but rather to the line in the function table containing up to six orders. These orders were fetched and executed in sequence unless a jump instruction was issued to transfer control to the first order of a new line. In addition to the range mentioned, addresses 000 to 007 had a special meaning: they addressed the eight fields in the Constant Transmitter that held the data read in from a punched card. This new feature allowed the execution of orders from punched cards in addition to executing orders from Function Tables. Once an address between 000 and 007 had been activated, orders were executed sequentially from the card and the address wrapped around to 000 automatically unless a jump instruction with a function table address was executed. Total program size thereby became unlimited: if the 4,800 order positions of the four function tables were not enough, additional orders could be executed from punched cards.

The use of the 20 accumulators became very specific with the implementation of the converter code. As I have mentioned, accumulator 6, used as the instruction counter, was subdivided into four subfields:

- 6 (11,10,9)—Used as an accumulator for signed 2-digit numbers.
- 6 (8,7)—Used as a 2-digit accumulator.
- 6 (6,5,4)—Used to hold the target address to be used by a conditional transfer order if the condition "positive" applied.
- 6 (3,2,1)—The actual instruction counter holding the current address, incremented by 1 automatically after the last order of the line specified had been executed.

Similarly, accumulator 8 was divided into two subfields:

- 8 (11, 4)—Could be used as storage.
- 8 (3,2,1)—Used as a table pointer used by the Function Table Number (FTN) and Function Table Constant (FTC) orders (see Table 2), which read numerical data from the function table. It was incremented by 1 automatically after each FTN or FTC order.

Accumulators 13 and 15 were used as a sort of scratch registers for various orders and as true accumulators adding any new value received to their present contents. Accumulators 1, 2, 3, 4, 5, 9, 10, 14, 16, 17, 18, 19, and 20 could be used to store intermediate results of calculations. The remaining accumulators—7, 11, and 12—were used for specific mathematical operations. Table 3 summarizes the use of the 20 accumulators as defined by the converter code.

Test program for the function tables

Clippinger provided a suggested test program for the ENIAC. To give readers a flavor of real ENIAC coding, the first part of the suggested test program is used as an example, which tests the read out capability from a function table. The first part testing the function tables occupies 12 lines in a function table. Clippinger merely gives the order codes of this program without further comments. Table 4 below lists the ENIAC program code together with the addresses at which the orders are stored in the first two columns as given by Clippinger in his report. The next three columns define the order used and provide an explanation of the effect the order has. The last column illustrates the major sequences of the program.

The program starts with an initialization sequence where variables and addresses are initialized. The main body of the program consists of two nested loops: an inner loop reading the same function table line one hundred times and accumulating the read-out values; and an outer loop repeating this procedure over 300 consecutive function table lines. Note that the small description Clippinger provided for this program part does not at all match with what the program actually does; there was also no further information in Clippinger's report to resolve this discrepancy. Clippinger's description is: "To test the function tables the number in the first row and first function table is sent to accumulator 11 or 15, 10**n times, then shifted and subtracted from the correct number in the constant transmitter; if the difference is not zero, it is printed and then the next number is sent to accumulator 11 or 15, . etc. The whole test takes about 15 minutes if n=3." (R.F. Clippinger, *A Logical Coding System Applied to the ENIAC*, BRL 673, report, Ballistic Research Laboratories, Aberdeen Proving Ground, 29 Sept. 1948; http://ftp.arl.army.mil/~mike/comphist/48eniac-coding/.)

There also seems to be an error contained in the program, because the accumulators used to accumulate the inner loop's read-out values are not reset in the outer loop. The effect is that they will continue to accumulate the read-out values over all 300 lines tested in the function table, an effect Clippinger could hardly have intended.

Table 1. List of initial ENIAC orders.*

Order Code	Order Name	Description of Order
00	C	Clear Accumulator 15.
01,02,03,04, 05, 10, 11, 12, 13, 14, 15, 21–25,	x[l]**	Accumulator x clear and receive from Accumulator 15, clear Accumulator 15 (x not equal to 6,13,15,20).
20	13[l]	Accumulator 13 hold and receive from Accumulator 15, clear Accumulator 15.
31–35,40–45, 51–54, 64	x[t]***	Accumulator x transmit to Accumulator 15 (x is not equal to 6,13,15,20).
50	13[t]	Accumulator 13 hold and transmit to Accumulator 15
30	X	Multiply Accumulator 11 by Accumulator 15, add product to Accumulator 13 and store in Accumulator 15.
55 55a	Divide	Divide Accumulator 15 by Accumulator 7, store quotient in Accumulator 15 and remainder in Accumulator 5.
60	Square root	Take square root of Accumulator 15 and store it in Accumulator 15, remainder in Accumulator 5.
63	M	Complement of Accumulator 15.
65	DS	Drop sign of the number in Accumulator 15.
90	Sh	Shift Accumulator 15. Direction of shift and number of places to be shifted indicated in second order position.
71	Halt	Do not get next order and cease computing.
80 81 82 83 84	AB CD EF GH JK	Clear Accumulator 11, then Constant Transmitter send first number to Accumulator 11 and second number to Accumulator 15.

Table 1. List of initial ENIAC orders (cont.).*

Order Code	Order Name	Description of Order
70	N2D	Send next pair of instruction digits to Accumulator 15.
72	F.T.	Clear Accumulator 11. Function Table transmit 12 digits and two signs from address specified in Accumulator 8. Accumulator 11 receive left sign and six digits, Accumulator 15 receive right sign and six digits. Increase address in Accumulator 8 by one.
61	Pr.	Move next card into printing position and print contents of Accumulators 1, 2, 15–20.
62	Rd.	Store information from card on relays of ENIAC reader.
75	C.T.	Conditional transfer to target address of Accumulator 6 if Accumulator 15 is positive.
85	Sh′	Shift Accumulator 15. Direction of shift and number of places to be shifted indicated in second order position. Place digits shifted out of Accumulator 15 into Accumulator 12.
93a	N3D8	Clear address subfield of Accumulator 8 and send next three instruction digits to this subfield.
94a	N3D6	Clear address subfield of Accumulator 6 and send next three instruction digits to this subfield.
95a	N6D6	Clear both address subfields of Accumulator 6 and send next six instruction digits to these subfields.
91a	20[l]	Accumulator 20 clear and receive from Accumulator 15, clear Accumulator 15.
92a	20[t]	Accumulator 20 transmit to Accumulator 15.
73a	6[l]	Accumulator 6 clear and receive from Accumulator 15, clear Accumulator 15.
74a	6[t]	Accumulator 6 transmit to Accumulator 15.
94	N4D	Send next four instruction digits to Accumulator 15.
95	N6D	Send next six instruction digits to Accumulator 15.
91	18 ↔ 20	Accumulators 18 and 20 interchange contents.
92	6(11,10,9)	Add sign and two digits from Accumulator 15 to Accumulator 6, store sum in Accumulator 15.
93	6(8,7)	Add two digits from Accumulator 15 to Accumulator 6, store sum in Accumulator 15.
73	6R3	Transmit address in Accumulator 15 to Accumulator 6, clear Accumulator 15.
74	6R6	Transmit two addresses in Accumulator 15 to Accumulator 6, clear Accumulator 15.

** R.F. Clippinger,* A Logical Coding System Applied to the ENIAC*, BRL 673, report, Ballistic Research Laboratories, Aberdeen Proving Ground, 29 Sept. 1948; http://ftp.arl.army.mil/~mike/comphist/48eniac-coding/.*

*** Clippinger's report explains that order 01 acts on accumulator 1, order 02 on acc. 2, 03 on 3, 04 on 4, 05 on 5, 10 on 7, 11 on 8, 12 on 9, 13 on 10, 14 on 11, 15 on 12, 21 on 14, 22 on 16, 23 on 17, 24 on 18, and 25 on 19.*

**** Clippinger's report explains that order 31 acts on accumulator 1, order 32 on acc. 2, 33 on 3, 34 on 4, 35 on 5, 40 on 7, 41 on 8, 42 on 9, 43 on 10, 44 on 11, 45 on 12, 51 on 14, 52 on 16, 53 on 17, 54 on 18, and 64 on 19.*

Table 2. List of ENIAC orders of the converter code.*

Order Code	Order Name	Description of Order
91	S.C.	Clears all accumulators except Accumulator 6.
15	CL	Clear Accumulator 15.
01,02,03,04,05, 07,08,09,10, 11,12,13,14,16, 17,18,19,20	XL**	Accumulator X clear and then receive from Accumulator 15, which transmits and then clears.
06	6L	Accumulator 6 clear and then receive from Accumulator 15, which transmits and then clears.
21,22,23,24,25, 26,27,28,29,30, 31,62,33,34,36, 37,38,39,40	XT***	Accumulator X transmit-hold to Accumulator 15.
92	6(11,10,9)	Add sign and two digits from Accumulator 15 to Accumulator 6, store sum in Accumulator 15.
93	6(8,7)	Add two digits from Accumulator 15 to Accumulator 6, store sum in Accumulator 15.
79	6(6,5,4)	Transfer address in Accumulator 6 clear and receive from Accumulator 15, which transmits and then clears.
32,43,42,53,52	SRX****	Shift digits in Accumulator 15 right X places.
60,71,70,81,80	SLX	Shift digits in Accumulator 15 left X places.
38,49,48,49,58	S′RX	Shift digits in Accumulator 15 right X places. Move digits shifted out to Accumulator 12.

Table 2. List of ENIAC orders of the converter code (cont.).*

Order Code	Order Name	Description of Order
66,77,76,87,86	S'LX	Shift digits in Accumulator left X places. Move digits shifted out to Accumulator 12.
41	M	Change sign and form tens complement of Accumulator 15.
46	D.S.	Drop sign in Accumulator 15.
82	A.V.	Take absolute value of Accumulator 15.
57	X	Multiply Accumulator 11 by Accumulator 15, add product to Accumulator 13 and store in Accumulator 15.
63	Divide	Divide Accumulator 15 by Accumulator 7, store quotient in Accumulator 15 and remainder in Accumulator 5.
64	Square root	Take square root of Accumulator 15 and store it in Accumulator 15, remainder in Accumulator 5.
45	Pr	Move next card into printing position and print contents of Accumulators 1, 2, 15–20.
44	Rd	Store information from card on relays of ENIAC reader.
75	N3D8	Clear address subfield of Accumulator 8 and send next three instruction digits to this subfield.
89	N3D6	Clear address subfield of Accumulator 6 and send next three instruction digits to this subfield.
47	FTN	Clear Accumulator 11. Function Table transmit 12 digits and two signs from address specified in Accumulator 8. Accumulator 11 receive left sign and six digits, Accumulator 15 receive right sign and six digits. Increase address in Accumulator 8 by one.
97	FTC	Function Table transmits first signed 10 digits from address specified in Accumulator 8 to Accumulator 15 adding to any contents of Accumulator 15.
50 51 54 55 56	AB CD EF GH JK	Clear Accumulator 11, then Constant Transmitter send first number to Accumulator 11 and second number to Accumulator 15.
72ab	N2D	Send next pair of instruction digits to Accumulator 15.
73abcd	N4D	Send next four instruction digits to Accumulator 15.
74abcdef	N6D	Send next six instruction digits to Accumulator 15.
78	6R3	Transmit address in Accumulator 15 to Accumulator 6, clear Accumulator 15.
83	N3D6	Clear address subfield of Accumulator 6 and send next three instruction digits to this subfield.
84	N6D6	Clear both address subfields of Accumulator 6 and send next six instruction digits to these subfields.
69	C.T.	Conditional transfer to target address of Accumulator 6 if Accumulator 15 is positive.
94	I	Count Order: This order enables the coder to perform in succession a series of not more than three distinct iterative processes where the induction variable counts are preset on the switches of the Master Programmer Unit.
95	di	Count Order: Skip the preset count.
96	Cdi	Count Order: Returns the count to the first count set in the machine.
90/99	D	Delay: Does nothing but go to the next order.
00/35	Halt	Do not get next order and cease computing.

** This table is based on the descriptions given by B.W. Fritz,* Description of the Eniac Converter Code, *BRLM 582, memorandum report, Ballistic Research Laboratories, Aberdeen Proving Ground, Dec. 1951.*

*** Fritz' report explains that order 01 acts on accumulator, order 02 on accumulator 2, and so on until order 20 that acts on accumulator 20.*

**** Fritz' report explains that order 21 acts on accumulator 1, order 22 on accumulator 2, 23 on 3, 24 on 4, 25 on 5, 26 on 6, 27 on 7, 28 on 8, 29 on 9, 30 on 10, 31 on 11, 62 on 12, 34 on 14, 36 on 16, 37 on 17, 38 on 18, 39 0n 19, and order 40 on accumulator 20.*

***** Fritz' report explains that order 32 shifts by 1 place, 43 by 2 places, and so on until order 52 shifts by 5 places. The same logic applies to the orders that follow: SLX, S'RX, S'LX.*

Table 3. Standard use of accumulators.

Accumulator Number	Accumulator Use
1	Output to Card Punch (1st number) , otherwise available as storage
2	Output to Card Punch (2nd number) , otherwise available as storage
3	Available as storage

Table 3. Standard use of accumulators (cont.).

Accumulator Number	Accumulator Use
4	Available as storage
5	Available as storage
6	Selects Function Table and Order (Instruction Counter)
7	Denominator (input for division)
8	Selects Function Table and Constant Number (Table Pointer)
9	Available as storage
10	Available as storage
11	Input from Constant Transmitter/Card Reader (first number) Input from Function Table (left number indicated in Accumulator 8) Multiplier (input for multiplication)
12	Stores Multiplicand after Multiplication Result of Division: Remainder Result of Square Root: Remainder Receives digits shifted out from Accumulator 15 with Shift Prime orders
13	Work Accumulator (Scratch Register) used for temporary storage during various orders
14	Available as storage
15	Input from Constant Transmitter/Card Reader (second number) Input from Function Table (order indicated in Accumulator 6) Input from Function Table (right number indicated in Accumulator 8) Output to Card Punch (third number) Multiplicand (input for multiplication) Result of Multiplication (11) * (15) Numerator (input for division) Result of Division: Quotient (15)/(7) Input for Square Root Result of Square Root Drop Sign Operation Shift Operation Number input for conditional jump on negative/positive
16	Output to Card Punch (4th number) , otherwise available as storage
17	Output to Card Punch (5th number) , otherwise available as storage
18	Output to Card Punch (6th number) , otherwise available as storage
19	Output to Card Punch (7th number) , otherwise available as storage
20	Output to Card Punch (8th number) , otherwise available as storage

Table 4. Test program for the function tables.*

Function Table Line	Order Code Digits	Order	Operation	Explanation	Program Sequence
00	94	N4D	Send next four digits to Accumulator 15 (4,3,2,1). Accumulator 15 (4,3,2,1) = XXX	XXX constitutes the address of a Function Table (FT) location to be read out.	Program Initialization
	0X				
	XX				
	11	8L	Accumulator 8 receives from Accumulator 15. Accumulator 8 (3,2,1) = XXX	XXX is moved to Accumulator 8 as address pointer to the FT.	
	00	C	Clear Accumulator 15.	Clear Accumulator 15.	
	00	C	Clear Accumulator 15.	No operation in this case.	
01	01	1L	Accumulator 1 receives from Accumulator 15.	Clears Accumulator 1.	

Table 4. Test program for the function tables.* (cont.)

Function Table Line	Order Code Digits	Order	Operation	Explanation	Program Sequence	
02	2L		Accumulator 2 receives from Accumulator 15.	Clears Accumulator 2.	Program Initialization	
	94	N4D	Send next four digits to Accumulator 15 (4,3,2,1). Accumulator 15 (4,3,2,1) = 100	Accumulator 15 is loaded with 100.		
	01					
	00					
	63	M	Form complement of Accumulator 15.	Accumulator 15 is loaded with –100.		
02	03	3L	Accumulator 3 receives from Accumulator 15.	Accumulator 3 is loaded with –100. → Accumulator 3 acts as iteration counter for the inner loop.		
	94	N4D	Send next 4 digits to Accumulator 15 (4,3,2,1). Accumulator 15 (4,3,2,1) = 300	Accumulator 15 is loaded with 300.		
	03					
	00					
	63	M	Form complement of Accumulator 15.	Accumulator 15 is loaded with –300.		
	04	4L	Accumulator 4 receives contents from Accumulator 15.	Accumulator 4 is loaded with –300. → Accumulator 4 acts as iteration counter for the outer loop		
03	70	N2D	Send next two digits to Accumulator 15 (2,1).	Accumulator 15 is loaded with 01.		
	01					
	63	M	Form complement of Accumulator 15.	Accumulator 15 is loaded with –01.		
	05	5L	Accumulator 5 receives from Accumulator 15.	Accumulator 5 is loaded with –01.		
	00	C	Clear Accumulator 15.	Clear Accumulator 15.		
	00	C	Clear Accumulator 15.	No operation in this case.		
04	95	N6D	Send next six digits to Accumulator 15 (6,5,4,3,2,1). Accumulator 15 (6,5,4,3,2,1) = (alpha)09(alpha)05	Accumulator 15 is loaded with '(alpha)09(alpha)05'. (alpha) identifies the FT used.	Inner Loop Initialization	Outer Loop
	(alpha)0					
	9(alpha)					
	05					
	74	6R6	Accumulator 6 receives from Accumulator 15. Unconditional jump to first order contained in line 05 of FT number Alpha.	Accumulator 6 is loaded with two addresses: Accumulator 6(6,5,4) = (alpha)09 → Transfer address Accumulator 6(3,2,1) = (alpha)05 → Next instruction address.		
	00	C	Clear Accumulator 15.	Placeholder, is not executed		
05	72	FT	Transmits 12 digits and two signs from FT to Accumulator 11 and 15. FT line number is taken from Accumulator 8 (3,2,1). Accumulator 8 (3,2,1) is increased by one at the end of the execution.	FT Line at address XXX is read. Left part and sign is moved to Accumulator 11 (10–5). Right part and sign is moved to Accumulator 15 (10–5).	Inner Iteration Loop	
	90	Sh	Shift contents of Accumulator 15 four places to right.	Contents of Accumulator 15 (10–5) are shifted to Accumulator 15 (6–1). Accumulator 15 now contains right part of FT XXX.		
	94					
	32	2T	Accumulator 2 sends its contents to Accumulator 15. Accumulator 15 adds the received number to its current contents.	Accumulator 2 contains the accumulated sum of the right part of FT XXX. This is added to the right part just read from FT XXX in Accumulator 15.		
	02	2L	Accumulator 2 receives contents from Accumulator 15.	Accumulator 15 moves its contents to Accumulator 2 and clears. → Accumulator 2 now contains the updated accumulated sum of the right part of FT XXX.		
	44	11T	Accumulator 11 sends its contents to Accumulator 15.	Accumulator 15 (10–5) now contains left part of FT XXX.		

Table 4. Test program for the function tables.* (cont.)

Function Table Line	Order Code Digits	Order	Operation	Explanation	Program Sequence: Inner Iteration Loop	Program Sequence: Outer Loop
06	90 94	Sh	Shift contents of Accumulator 15 four places to right.	Contents of Accumulator 15 (10–5) are shifted to Accumulator 15 (6–1). Accumulator 15 now contains left part of FT XXX.		
	31	2T	Accumulator 1 sends its contents to Accumulator 15. Accumulator 15 adds the received number to its current contents.	Accumulator 1 contains the accumulated sum of the left part of FT XXX. This is added to the left part just read from FT XXX in Accumulator 15.		
	01	1L	Accumulator 1 receives contents from Accumulator 15.	Accumulator 15 moves its contents to Accumulator 1 and clears. → Accumulator 1 now contains the updated accumulated sum of the left part of FT XXX.		
	41	8T	Accumulator 8 sends its contents to Accumulator 15.	Accumulator 15 now contains the address of the next line of the FT to be read out.		
	35	5T	Accumulator 5 sends its contents to Accumulator 15. Accumulator 15 adds the received number to its current contents.	Accumulator 5 contains –1, which is added to Accumulator 15. → 1 is subtracted from the FT address, resulting in the original address XXX in Accumulator 15.		
07	11	8L	Accumulator 8 receives from Accumulator 15. Accumulator 15 clears at the end.	Original address is restored in Accumulator 8.		
	35	5T	Accumulator 5 sends its contents to Accumulator 15.	Accumulator 5 still contains –1. → Accumulator 15 now contains –1.		
	63	M	Form complement of Accumulator 15.	Accumulator 15 now contains +1.		
	33	3T	Accumulator 3 sends its contents to Accumulator 15. Accumulator 15 adds the received number to its current contents	Decrease iteration counter of inner loop by 1. (Iteration count is negative, 1 is added to decrease it.)		
	03	3L	Accumulator 3 clears and receives from Accumulator 15. Accumulator 15 clears at the end.	Accumulator 3 now contains decreased iteration count for inner loop.		
	33	3T	Accumulator 3 sends its contents to Accumulator 15. Accumulator 15 adds the received number to its current contents.	Decreased iteration count is moved to Accumulator 15 to be tested by conditional jump instruction that follows next.		
08	75	C.T.	Conditional jump instruction: 1) If Accumulator 15 is negative, continue with next order. 2) If Accumulator 15 is zero or positive, jump to transfer address.	1) If iteration count is still negative, execute next order. → Continue iteration of inner loop. 2) If iteration count has reached zero, jump to transfer address. → Stop iteration of inner loop and continue with first order of outer loop at line 09.		
	94 0(alpha) 05	N4D	Send next four digits to Accumulator 15 (4,3,2,1). Accumulator 15 (4,3,2,1) = 0(alpha)05	Accumulator 15 (3,2,1) is loaded with (alpha) 05. → Start address of iteration loop.		
	73	6R3	Address in Accumulator 15 (3,2,1) is moved to Accumulator 6 (3,2,1). Next order at Accumulator 6 (3,2,1) is executed.	Unconditional jump to the beginning of the iteration loop.		
	00	C	Clear Accumulator 15.	No operation in this case. End of inner loop.		

Table 4. Test program for the function tables.* (cont.)

Function Table Line	Order Code Digits	Order	Operation	Explanation	Program Sequence	
09	41	8T	Accumulator 8 sends its contents to Accumulator 15.	Accumulator 15 now contains the address of the last line of the FT read out.		Outer Loop
	61	Pr	Punch contents of Accumulator 1,2, 15–20. Accumulators are not cleared and hold their contents.	Output of the accumulated sums of the left and of the right part of the read outs from the FT Lines and of the address of the last read FT line.		
	70 01	N2D	Send next two digits 01 to Accumulator 15 (2,1).	Accumulator 15 adds 01 to its present contents. → Accumulator 15 now contains the address of the next sequential FT line to be read.		
	11	8L	Accumulator 8 receives contents from Accumulator 15. Accumulator 15 clears at the end.	Accumulator 8 now contains the increased address of the next FT line to be read out.		
	35	5T	Accumulator 5 sends to Accumulator 15.	Accumulator 15 contains –1.		
10	90 02	Sh	Shift contents of Accumulator 15 2 places to left.	Accumulator 15 contains –100.		
	03	3L	Accumulator 3 receives contents from Accumulator 15. Accumulator 15 clears at the end.	Accumulator 3 contains –100. → the iteration count for the inner loop is restored.		
	70 01	N2D	Send next two digits to Accumulator 15(2,1). Accumulator 15(2,1) = '01'	Accumulator 15 is loaded with 01.		
	34	4T	Accumulator 4 sends its contents to Accumulator 15. Accumulator 15 adds the received number to its current contents.	Decrease iteration counter of outer loop by 1. (Iteration count is negative, 1 is added to decrease it.)		
11	04	4L	Accumulator 4 receives contents from Accumulator 15. Accumulator 15 clears at the end.	Accumulator 4 now contains decreased iteration count for outer loop.		
	95 (alpha)1 3(alpha) 12	N6D	Send next six digits to Accumulator 15 (6,5,4,3,2,1). Accumulator 15 (6,5,4,3,2,1) = (alpha)13(alpha)12.	Accumulator 15 is loaded with (alpha)13(alpha)12		
	74	6R6	Accumulator 6 receives contents from Accumulator 15. Accumulator 15 clears. Unconditional jump to first order contained in line 05 of FT number Alpha.	Accumulator 6 is loaded with two addresses: Accumulator 6 (6,5,4) = (alpha)13 → Address of next test. Accumulator 6 (3,2,1) = (alpha)12 → Next instruction address.		
12	34	4T	Accumulator 4 sends its contents to Accumulator 15.	Decreased iteration count is moved to Accumulator 15 to be tested by conditional jump instruction that follows next.		
	75	C.T.	Conditional jump instruction: 1) If Accumulator 15 is negative, continue with next order. 2) If Accumulator 15 is zero or positive, jump to transfer address.	If iteration count is still negative, execute next order. → Continue iteration of outer loop. If iteration count has reached zero, jump to transfer address. → Stop outer loop and continue with next test at first order of line 13.		
	94 0(alpha) 04	N4D	Send next four digits to Accumulator 15 (4,3,2,1). Accumulator 15 (4,3,2,1) = '0(alpha)04'	Accumulator 15 (3,2,1) is loaded with (alpha)04. → Jump to begin of outer loop.		
	74	6R6	Accumulator 6 receives contents from Accumulator 15. Unconditional jump to first order contained 04 of FT number Alpha.	Accumulator 6 is loaded with two addresses: Accumulator 6 (6,5,4) = 000 → Dummy transfer address. Accumulator 6 (3,2,1) = (alpha)04 → First order of outer loop.		

** This table is based on the descriptions given by B.W. Fritz,* Description of the Eniac Converter Code, *BRLM 582, memorandum report, Ballistic Research Laboratories, Aberdeen Proving Ground, Dec. 1951.*