Burroughs TC 500

OPERATION AND PROGRAMING MANUAL

Part III

TC 500 BASIC ASSEMBLER LANGUAGE

Printed in U. S. America February, 1969 For 1034006

TABLE OF CONTENTS

PART III BASIC ASSEMBLY LANGUAGE – TC 500

PART III - BASIC ASSEMBLER LANGUAGE

1. GENERAL DESCRIPTION

The TC 500 Basic Assembler Language is a symbolic programing language for writing programs which are to be run on TC 500 Systems using GP 300 Series Firmware. Programs written in this language may be assembled on a TC 500, a B 2500/B 3500, or a B 5500 System.

All valid GP 300 Series Instructions are listed including the permissible numeric and/or symbolic parameter(s) for each (refer to Part II, "General Purpose Language 300" for a detailed explanation of functional results). In addition, all Basic Assembler Pseudo Instructions are described and the use of the Assembler Coding form is explained.

2. ASSEMBLY CODING FORM

TC 500 programs, which are to be assembled with the Basic Assembler Program, are written on the Assembler Coding Form to provide the proper documentation for either keyboard input or for the creation of a punched card input deck.

The Assembly Coding Form is graphically described in this section. When the program data is to be punched into cards for punched card input during assembly, it is necessary that the data be justified right or left within the field on the coding form as specified for each entry below:

2.1 LIBRARY CODE – COLUMN 1

When modifying Library Routines, enter any alpha character; otherwise, leave blank. The characters "C" for "Change", "A" for "Add", or "D" for "Delete" are recommended to provide uniform interpretation.

2.2 PROGRAM ID – COLUMNS 5-10

Enter Program Identification in this field. Justify left or right.

2.3 SEQUENCE NUMBER – COLUMNS 11-15 (PUNCHED CARD SOURCE PROGRAM ONLY)

On Keyboard or Paper Tape source programs, the Basic Assembler numbers each line of coding in ascending sequence. On Punched Card source programs, the user may enter pre-assigned sequence numbers, right justified, or may have the assembler automatically assign numbers in ascending sequence.

2.4 LABEL – COLUMNS 16-21

A label is a symbolic designation to describe a parameter for a memory location or some other parameter value. It is commonly used to identify the starting word and syllable of a subroutine, to identify a word or area of memory for accumulations of data or storage of constant data, or to identify a particular line number or printing position on an output form. The Assembler Language Description indicates which GP 300 instructions and Assembler Pseudo Instructions may use, must use, or must not use a label as the parameter (see section 3).

A label entry in this field (columns 16 to 21) denotes that this line of code will define the purpose or value of the label. Or, in the case of a labeled subroutine, this line is the first line of code in that subroutine. A label may be used in a parameter before or following its use in this field. However, each unique label used as a parameter must eventually be defined by use in this field.

A label may consist of from one (1) to six (6) alphanumeric characters. It must be left justified and the first character must be an alpha character. A label may be the same as the mnemonic operation code of any GP 300 instruction or Assembler pseudo instruction. However, such use of op codes for labels could lead to confusion in program interpretation.

2.5 OP CODE – COLUMNS 22-26

The applicable instruction or Pseudo Instruction is entered in this field, left justified.

2.6 FLD LN (FIELD LENGTH) COLUMNS 27-28 (PUNCHED CARD SOURCE PROGRAMS ONLY)

An entry is made here for the Pseudo Instructions ALF and MASK to indicate the number of characters or digits in the constant which start in column 29. Entry is right justified.

The code "CC" is entered in columns 27-28 when a continuation card is necessary (an alpha constant or print mask exceeds 24 characters).

2.7 A – PARAMETER, LABEL – COLUMNS 29-34

The applicable Label or Parameter is entered in this field. Parameter entries are left justified. Label entries are left justified, may consist of from one (1) to six (6) alphanumeric characters, and the first character must be an alpha character. A Label used with the Pseudo Instruction "LIB" may contain up to ten (10) characters in which case the field would include columns 29 to 38 (refer to 2.4 for a discussion of labels).

2.8 A – PARAMETER, ± INC / REL – COLUMNS 35-38

A signed numeric entry may be made in this field to denote a plus or minus value for incrementing or relative addressing with the label in columns 29-34 as the base; or if a label is not used, the syllable location of this instruction is the base. The maximum increment is 255.

If the entry is a negative value, the "-" must be entered in column 35. If the entry is plus, the "+" is not entered. The value is entered in columns 36-38 right justified.

2.9 B – PARAMETER – COLUMNS 39-42

The applicable alphanumeric parameter is entered in this field, left justified.

2.10 C – PARAMETER – COLUMN 43

The applicable numeric parameter is entered in this column.

2.11 ALPHANUMERIC DATA (ALPHA CONSTANT AND PRINT MASK) – COLUMNS 29-52

Alphanumeric data is entered here for the Pseudo Instructions "ALF" and "MASK", left justified. If an alpha constant or print mask is more than 24 characters in length, those characters in excess of 24 are continued on the next line of coding starting in column 29 and preceded by "CC" (Continuation Card) in columns 27-28. The continuation card(s) also must contain the Pseudo Instruction and a sequence number.

2.12 CONSTANT DATA (NUMERIC CONSTANT) – COLUMNS 29-47

Numeric Constant data is entered here for the Pseudo Instruction "NUM", left justified. The flags "-", "C", "M" must be entered to the left of the most significant digit in the numeric value.
Example: "-C54251"

2.13 REMARKS – COLUMNS 53-77

Remarks may be entered in this field, and will appear in the printed documentation. Remarks that are entered with the Pseudo Instruction "DOC" may start in column 29.

2.14 SUMMARY OF ENTRIES ON ASSEMBLER CODING FORM

DESCRIPTION	COLUMNS	JUSTIFIED
LIBRARY CODE	1	
NOT USED	2-4	

DESCRIPTION	COLUMNS	JUSTIFIED
PROGRAM I.D.	5-10	LEFT OR RIGHT
SEQUENCE NO.	11-15	RIGHT
LABEL	16-21	LEFT
OP. CODE	22-26	LEFT
FIELD LENGTH OR "CC"	27-28	RIGHT
A PARAMETER	29-34	LEFT
± INC OR REL	35-38	RIGHT *
B PARAMETER	39-42	LEFT
C PARAMETER	43	
NUMERIC CONSTANT	29-47	LEFT
ALPHA CONSTANT OR MASK	29-52	LEFT
REMARKS	53-77	LEFT
NOT USED	78-80	

* NOTE: If the ± INC or REL is a negative the "-" must be shown in column 35.

Example:

-		1	4
35	36	37	38

2.15 EXAMPLE OF ASSEMBLER LANGUAGE CODING

The sample program following illustrates the proper use of the coding form and the Assembler Pseudo Instructions.

BURROUGHS ASSEMBLER CODING FORM

PROGRAM I D: 15

PAGE 1 OF 2

ADDING MACHINE PROGRAM

CODE	SEQUENCE	LABEL	OPERATION CODE	FIELD LENGTH	PARAMETER A LABEL	PARAMETER A + OR - INC / REL	PARAMETER B	PARAMETER C	REMARKS
	1		NØTE						ADDING MACHINE
	2		NØTE						ØNE TOTAL ØF 15 DIGITS
	3		LPNR		PMASK				
	4		LPKR		PKEYS				
	5	START	CLM		TØTAL				
	6		AL		1				
	7		PØS		LIST				
	8		PKA		12				TØ SUBTØTAL AND TØTAL
	9		NKRCM		AMØUNT				LIST ØCK 1 = AMØUNT
	10		PNS-		PRINT				
	11		SK		K		2	2	ØCK 2 = NUMBER
	12		ADM		TØTAL				ADD TØ TØTAL
	13		PC-		-				
	14		EX		K		2	1	TEST IF NUMBER
	15		PCP		#				
	16		BRU		START	1			
	17	PRTSUB	SRJ		CØMR				PRINT SUBTØTAL
	18		PC-		C				
	19		PC+		S				
	20		BRU		START	1			

CONSTANT DATA (NUMERIC)

ALPHANUMERIC DATA OR PRINT MASK

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 20 24

BURROUGHS ASSEMBLER CODING FORM

PROGRAM I D: 15

PAGE 2 OF 2

ADDING MACHINE PROGRAM

CODE	SEQUENCE	LABEL	OPERATION CODE	FIELD LENGTH	PARAMETER A LABEL	PARAMETER A + OR - INC / REL	PARAMETER B	PARAMETER C	REMARKS
	21	PRTØT	SRJ		CØMR				PRINT TØTAL
	22		PC-		CR				
	23		PC+		*				
	24		BRU		START				
	25	CØMR	TRA		TØTAL				CØMMØN TØTAL RØUTINE
	26		PNS-		PRINT				
	27		PØS		SYMBØL				
	28		SRR		1				
	29		WØRD						
	30	PKEYS	BRU		PRTSUB				PKA 1 SUBTØTAL
	31		BRU		PRTØT				PKA 2 TØTAL
	32	PMASK	MASK	20	Z,ZZZ,ZZZ,ZZZ,ZZZ.DD				
	33	TØTAL	REG		1				
	34	AMØUNT	DEFT		15		0		
	35	PRINT	DEFT		14		0		
	36	LIST	DEF		10				
	37	SYMBØL	DEF		32				
	38		END						

CONSTANT DATA (NUMERIC)

ALPHANUMERIC DATA OR PRINT MASK

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 20

3. LANGUAGE DESCRIPTION

This section describes the Assembler Pseudo Instructions and the proper form for the GP 300 Instructions in Assembler programing.

3.1 PSEUDO INSTRUCTIONS

Pseudo instructions control the manner of assembly and determine the interpretation of data fed to the assembler. They generally do not directly produce machine language instructions, except in some cases where they fill in syllables to increment the program counter to the next word.

The following instructions are valid for this Basic Assembler Language.

3.1.01 Origin Instruction

Op Code	A	B
ORG	0:N	

The first instruction following the Origin Pseudo-instruction will be assembled in Syllable 0 of the word specified in the parameter field. If the specified word has already been assigned by the assembler, an error message will be printed and entry assignment will start at the same sequence number. No machine language instruction is assembled.

3.1.02 Region Instruction

Op Code	A	B
REG	1:255	

The Pseudo - Instruction "REG" would be used when one or more words of memory must be set aside for use for Accumulating Totals or merely as working storage.

If the syllable counter is not "0" it is incremented and "Stop" instructions inserted until the counter reaches "0".

The word counter is advanced by the number in the parameter field.

If the word counter exceeds "n" after advancement, an error message is printed and entry assignment will start at the same sequence number. No machine language instruction is assembled.

The region must be identified by placing its name label in the label field (columns 16 to 21) of the coding form.

3.1.03 Note Instruction

Op Code	A	B
NOTE		

This pseudo instruction will permit the entry of up to 25 characters in the "Remarks" field (columns 53 to 77). No machine language instruction is assembled. No Parameter field entry is required. If one is given, it will be ignored.

3.1.04 Page Instruction

Op Code	A	B
PAGE		

This pseudo instruction advances the assembler output form to align with the first line of the next page. No Parameter field entry is required. If one is given, it will be ignored. No machine language instruction is assembled.

3.1.05 Word Instruction

Op Code	A	B
WORD		

This pseudo instruction causes the assembler to assign the next instruction starting at the beginning syllable of the next word.

If the syllable counter is not "0", it will be incremented and Stop instructions inserted into each syllable until the counter reaches "0".

This instruction should immediately precede the entry of a Program Key Table.

3.1.06 Advance Line Instruction

Op Code	A	B
ADVL	1:4	

This pseudo instruction will advance the assembler output form the number of lines specified in the Parameter field. No machine language instruction is assembled.

3.1.07 Number Instruction

Op Code	A	B
NUM		

The NUM pseudo instruction permits a word of numeric data to be stored as constant data in memory during program loading. Thus, this instruction can be used to clear a word by setting its value to zero.

A numeric constant of from 0 to 15 digits consisting of the digits 0-9 is accepted. In addition, the "-", "C" and "M" codes, preceding the digit position of the constant are accepted, and set their respective flags in the flag position of the word.

If the syllable counter is not "0", "Stop" instructions are inserted until the counter is "0". The numeric constant is then assembled in the next full word, right justified.

The number must be identified by placing its name label in the label field (columns 16 to 21) of the coding form, unless reference to it will be made by + or - incrementing from another entry.

3.1.08 Alpha Instruction

Op Code	A	B
ALF		

The ALF pseudo instruction permits alphanumeric data to be stored as constant data in memory during program loading.

An alphanumeric constant of up to 24 characters is accepted. Any character on the keyboard, including space, is valid. If more than 24 characters are required, a second line of code is used to continue the alpha entry and is preceded with "CC" in columns 27 and 28.

If the syllable counter is not "0" at the beginning of the "ALF" instruction, "Stop" instructions are inserted until the counter is "0". The alphanumeric constant is then assembled starting the next full word.

The alpha data must be identified by placing its name label in the label field (columns 16 to 21) of the coding form, unless reference to it will be made by + or - incrementing from another entry.

3.1.09 Mask Instruction

Op Code	A	B
MASK		

The MASK pseudo instruction is used to enter the table of mask words. An entry of up to 24 print format characters is accepted. These characters are as follows:

Group 1	Group 2				
F	I	.D	Z	C	.X
+	E	D:	Z:	.C	S
P	D	D,	Z,	X	

The Group 1 characters are valid only if they precede the Group 2 characters, otherwise they are ignored.

If the syllable counter is not "0" at the beginning of the MASK instruction, "Stop" instructions are inserted until the counter reaches "0". The Mask characters are then assembled in the next full word.

The appearance of any character other than those listed in Group 1 or Group 2 will cause an error condition to occur. Entry restarts at the same sequence number.

The mask table must be identified by placing its name label in the label field (columns 16 to 21) on the line of the first mask word entry.

3.1.10 Define Instructions

Op Code	A	B
DEF	0:N	
DEFT	0:15	0:15

The DEF or DEFT pseudo instructions may be used to define a label (i.e., assign a numeric value to it). This applies to labels that name something other than a memory location, and that represent a numeric parameter(s).

If DEF is entered, a numeric parameter "A" field entry of 0:N is accepted.

Ex:	Label	Op Code	A	B
		TK	NAME	
		_____		
		_____		
		_____		
	NAME	DEF	25	

If DEFT is entered, an "A" and "B" field are both accepted, permitting a numeric parameter of 0:15 in each.

Ex:	Label	Op Code	A	B
		NK	ORDER#	
		_____		
		_____		
		_____		
	ORDER#	DEFT	6	0

A symbolic label must be placed in the Label field (columns 16 to 21) of the coding form for each DEF or DEFT used, to designate the thing that is being defined.

3.1.11 Equate Instruction

Op Code	A	B
EQU		

This pseudo instruction will permit one label to be given the identical value of another label. The label coded in columns 16 to 21 will be equated to the label in columns 29 to 34. The parameter field label (columns 29 to 34) must have been previously used or defined.

3.1.12 End Instruction

Op Code	A	B
END		

The END pseudo instruction terminates the assembly program and must be used as the last line of code in the program.

3.1.13 Library Instruction (Used only for assembly on B 2500/B 3500, B 5500)

Op Code	A	B
LIB	Label	

The use of the Library instruction (LIB) signifies a subroutine to be inserted in the program at this point. The subroutine is identified by a label of up to 10 characters, and is contained in a file which is accessible to the B 3500.

Library Routines inserted into a program may be modified in the following manner: Any alphanumeric character in card column 1 on lines of code immediately following a LIB instruction will cause modification of a line of code in that library routine. As many changes as desired may be made to the routine so long as they follow the LIB instruction in succession. (All desired changes must be made to a LIB routine before continuing with further programing).

For convenience in interpretation, the following alphanumeric characters are recommended as codes for use in column one: "C" = Change, "A" = Add, "D" = Delete. The sequence number identifies which subroutine instruction is being modified. Label, Operation Code, and Parameters are entered according to the resulting change (or addition) desired. For a deletion, it is only necessary to enter the sequence number. The individual instructions contained in each library routine are documented in the library directory.

EXAMPLES:

	SEQ.#	Op Code	Parameters	Remarks
	00150	LIB	T101230400	(Insert Routine)
C	00020	POS	26	(Change Position Location)
A	00021	TK	15	(Include Typing)
D	00050			(Delete this Instruction)

A library subroutine cannot be specified to be placed within another library subroutine; that is, subroutines cannot be "nested" within other subroutines.

3.1.14 Documentation Instruction (Used only for assembly on B 2500/B 3500, B 5500)

Op Code	A	B
DOC		

The use of the "DOC" instruction permits more extensive narrative to be included in programs and in the subroutine library. Remarks of up to 49 characters are entered (starting in card column 29) which print on the assembly documentation from the B 3500, but which do not punch into the

symbolic program tape (or card deck). Thus, each complete assembly on the B 3500 using the source input deck provides this documentation; any re-assembly on a TC 500 would not provide it.

3.1.15 Establish Buffer Instruction

Op Code	A	B
ESTB		

The pseudo instruction ESTB is used for reserving main memory buffer areas in connection with the Data Communication message handling instructions. This is required when it is desired to move a message from the Data Communication Message Received Buffer into main memory before unpacking the message, or to build a message in main memory and then transfer it (completely formatted) to the Data Communication Transmit Buffer.

The ESTB instruction reserves a 32 word area (256 characters) or 1 track in user memory. It selects the highest track of user memory that is available, reserving 32 words starting with the 1st word of that track. For example, if 384 words of user memory (0 to 383) are designated in the program assembly, the 1st use of ESTB would reserve words 352 through 383; the 2nd use of ESTB would reserve words 320 through 351.

ESTB has no parameters, but it must be labeled. For example, if main memory record areas are to be reserved for messages received and messages to be transmitted, they might be labeled "RECEIV" and "SEND", and would be reserved in this manner:

Label	Op Code	A	B	C
RECEIV	ESTB			
SEND	ESTB			

In the above example, "RECEIV" would be assembled with a word number of 352 and "SEND" would be assembled with a word number of 320.

The subsequent transfer of data between these areas and the Data Communication processor buffers would be programed thus:

	Label	Op Code	A
Receive buffer to main memory:		TRB	RECEIV
or			
Main memory to transmit buffer:		TSB	SEND

The memory word number assembled for "RECEIV" and "SEND" is converted to a track number for TRB and TSB by the assembler. The word number prints to the right of the symbolic label on the listing; the track number appears in the machine code.

The unpacking or building of a message in one of these main memory record areas must be preceded by identifying the memory location with "LRBR" or "LKBR":

	Label	Op Code	A
		LRBR	RECEIV
or		LKBR	SEND

The unpacking or building of a message directly from or into the Data Communication processor buffers is pre-designated with LRBR or LKBR without using a label:

	Label	Op Code	A
Designate receive buffer		LRBR	
or			
Designate send buffer		LKBR	

Note: If the last user word is specified in assembly rather than the total number of words of user memory (ex: 383 rather than 384) the ESTB instruction will select the next lower track (ex: 320 to 352). This would cause the last 32 words to be inaccessable to the assembler for other use.

3.1.16 CODE Instruction

Op Code	A	B
CODE	(4 hexadecimal digits)	

The pseudo instruction CODE permits the insertion of 4 hexadecimal digits into a syllable of a word of memory. The value designated by the 4 digits in the A parameter is assembled into the next syllable of memory available. Other instructions may precede or follow its use in the same word of memory, or it may be used successively to assemble a full word or several words.

This instruction is useful in special situations, such as:

a) Inserting constant data consisting of USASCII codes (other than the 64 graphic characters which can be entered through the keyboard with the pseudo instruction "ALF"). Each such USASCII code is represented by 2 hexadecimal digits corresponding to the column and row number of the code in the USASCII table.

b) Packing a word of memory to include both alpha characters and numeric digits. The example below illustrates an alpha constant "SAME" and a numeric account number constant "105331" ("SAME" requires 10 digits; the account number only requires 6 digits):

	S	A	M	E	NUL	105331
Data required						
Required value in word:	5 3	4 1	4 D	4 5	0 0	105331

	Label	Op Code	A
		WORD	
(Syl. 0)	ACCT#	CODE	5331
(Syl. 1)		CODE	0010
(Syl. 2)		CODE	4D45
(Syl. 3)	SAME	CODE	5341

c) Assembling an overlay program which must tie into and use segments of another program. For example, the location of the "DATECG" routine, which is referred to in the "SRJ" instruction below, is not known by the assembler in this assembly:

Label	Op Code	A	B
PKEYS4	SRJ	DATECG	("DATECG" is a routine in another program.)

This can be assembled correctly by inserting the actual memory location with "CODE". The actual machine code can normally be determined by referring to the program listing in which the routine is used. Otherwise, the machine code can be obtained from the Memory Modify – Machine Code Tables:

Label	Op Code	A	B
PKEYS4	CODE	28EC	(SRJ to syllable 2 of word 236)

3.2 GP 300 SERIES FIRMWARE INSTRUCTION LIST

All of the GP 300 Instructions which are valid for the Basic Assembly Language are listed in the following paragraphs.

The form of Parameter field entry will vary by the type of instruction. The following charts indicate the acceptable parameter field entries. In some cases either a numeric or symbolic parameter field may be entered. These are shown with entries in both numeric and symbolic columns on the charts. If only one type of parameter is shown, only that type may be used.

3.2.01 Numeric Keyboard

GP 300 SERIES FIRMWARE INSTRUCTIONS

PARAMETERS

Part II Paragraph No.		INSTR.	NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
2.2.01	Numeric Keyboard	NK	0:15	0:15		Label			A = Max Digits Left of Dec. Pt. B = Max Digits Right of Dec. Pt.
2.2.01	Numeric Kbd., Permit Reverse Entry Key	NKR	0:15	0:15		Label			
2.2.01	Numeric Kbd., Permit C & M Keys	NKCM	0:15	0:15		Label			
2.2.01	Numeric Kbd., Permit Reverse Entry, C & M Keys	NKRCM	0:15	0:15		Label			

3.2.02 Alpha Keyboard

Part II Paragraph No.		INSTR.	NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
2.3.02	Type	TK	0:150			Label (1)			A = Max No. of Characters allowed
2.3.03	Type into Memory, Print	TKM	0:150			Label (1)			
2.3.04	Enter Alpha into Memory, Non-Print	EAM	0:150			Label (1)			
2.3.06	Load Keyboard Base Register	LKBR				Label (2)			
2.5.02	Enable Program Key Group A	PKA	1234 5678						Gr. A, 1-8 = First through eighth keys; Gr. B, 1-8 = Ninth through sixteenth keys; counting left to right
2.5.02	Enable Program Key Group B	PKB	1234 5678						
2.5.03	Load Program Key Base Register	LPKR				Label (2)			(1) May have + or - increment value (2) Memory location; May have + or - Relative Addressing

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
3.4.05	Load Print Numeric Base Register	LPNR				Label (2)			
3.4.01	Print Numeric	PN	0:14	0:15		Label			A = Pointer
3.4.01	Print Numeric, Shift Ribbon if Minus	PNS-	0:14	0:15		Label			B = Print Mask Address Modifier
3.4.01	Print Numeric, Shift Ribbon if Plus	PNS+	0:14	0:15		Label			
3.5.01	Print Character	PC	Char						
3.5.01	Print Character, Previous Ribbon	PCP	Char						
3.5.02	Print Character, if Accumulator Minus, Previous Ribbon	PC-	Char						(1) May have + or - increment value
3.5.02	Print Character, if Accumulator Plus, Previous Ribbon	PC+	Char						(2) Memory location; May have + or - Relative Address-
3.3.01	Print Alphanumeric	PA				Label (2)			
3.1.01	Load Position Register	POS	1:150**			Label (1)			** A = Actual Print Position
3.2.01	Red Ribbon	RR							

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	B	C	SYMBOLIC A	B	C	REMARKS
4.1.01	Open Forms Transport	OC	0:255			Label (1)			Advance 'A' Lines After Close
4.1.02	Close Forms Transport	CC							
4.2.01	Load Left Platen Count Register	LLCR	0:255			Label (1)			Load Register with "A" Value
4.2.01	Load Right Platen Count Register	LRCR	0:255			Label (1)			
4.2.01	Load Left Platen Limit Register	LLLR	0:255			Label (1)			
4.2.01	Load Right Platen Limit Register	LRLR	0:255			Label (1)			
4.3.01	Advance Left Platen	AL	0:255			Label (1)			Advance "A" Lines
4.3.01	Advance Right Platen	AR	0:255			Label (1)			
4.3.01	Advance Both Platens	ALR	0:255			Label (1)			
4.3.01	Advance Left Platen To	ALTO	1:255			Label (1)			Advance to Line "A"
4.3.01	Advance Right Platen To	ARTO	1:255			Label (1)			

(1) May have + or - increment value

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
5.1.01	Add to Accumulator	ADA				Label (2)			(1) May have + or - increment value
5.1.01	Add to Memory	ADM				Label (2)			
5.1.02	Subtract from Accumulator	SUA				Label (2)			(2) Memory location; May have + or - Relative Addressing
5.1.02	Subtract from Memory	SUM				Label (2)			
5.2	Transfer to Accumulator	TRA				Label (2)			
5.2	Transfer to Memory	TRM				Label (2)			
5.3	Clear Memory Word	CLM				Label (2)			
5.3	Clear Accumulator, Insert Constant	CLA	0:15	0:15		Label			A = Accum. Digit Pos. B = Constant
5.4.01	Insert Constant in Accumulator	INK	0:15	0:15		Label			
5.4.02	Add Constant to Accumulator	ADK	0:14	0:9					
5.4.03	Subtract Constant from Accumulator	SUK	0:14	0:9					
5.5.01	Load Shift Register	LSR	0:15			Label (1)			
5.5.02	Multiply	MUL				Label (2)			
5.5.02	Multiply and Round	MULR				Label (2)			
5.5.03	Divide	DIV				Label (2)			
5.5.05	Transfer Remainder to Accumulator	REM							Right Justified in Accumulator
5.6.01	Shift Off	SLRO	0:14	0:14		Label			A = No. of Digit Positions left B = No. of Digit Positions right
5.6.02	Shift Off with Sign	SLROS	0:15	0:15		Label			

3.2.06 Flags

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
6.7.01	Load Flags	LOD	A,K,R, P,X,Y	1234 -SCM					Accumulator Flags (A) = -SCM
6.7.02	Set Flags	SET	A,K,R, P,X,Y	1234 -SCM					OCK Flags (K) = 1234
6.7.03	Reset Flags	RST	A,K,R, P,X,Y	1234 -SCM					Reader Flags (R) = 1234
6.7.04	Change Flags	CHG	A,K,R, P,X,Y	1234 -SCM					Punch Flags (P) = 1234
									X Flags (X) = 1234
									Y Flags (Y) = 1234
									- = Sign
									S = Special
									C = Per Hundred
									M = Per Thousand

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
7.1.01	Load Index Register	LIR	1:4	0:255					A = Index Register B = Constant
7.1.02	Increment Index Register	IIR	1:4	0:255					A = Index Register B = Limit
7.1.03	Decrement Index Register	DIR	1:4	0:255					
7.1.04	Add to Index Register	ADIR	1:4	0:255					A = Index Register B = Constant
7.1.05	Transfer Accumulator to Index Register	TAIR	1:4						A = Index Register
7.2.01	Modify by Index Register	MOD	1:4						

3.2.08 Branch

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
8.1	Branch Unconditional	BRU				Label (4)			(4) Word and Syllable location; May have + or - Relative Addressing with label; or Label Field may be empty if + or - Relative Addressing used. In this case the location of this instruction is assumed as the label. Relative Addressing is in terms of syllables.
8.2.01	Subroutine Jump	SRJ				Label (3)			
8.2.02	Subroutine Return	SRR	1:4						A = Return Level (3) Word and Syllable location; May have + or - Relative Addressing in terms of syllables.

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	B*	C	SYMBOLIC A	B	C	REMARKS
9.1.01	Skip If Any Flags	SK	A,T,K, P,R,X, Y	-SCM 1234 OLIU	1:4				C = No. of Instructions
9.1.01	Skip If Every Flag	SKE	A,T,K, P,R,X, Y	-SCM 1234 OLIU	1:4				*O = Overflow L = Forms Limit I = Index Register U = Unassigned
9.1.02	Execute If Any Flags	EX	A,T,K, P,R,X, Y	-SCM 1234 OLIU	1:4				- = Sign S = Special C = Per Hundred M = Per Thousand
9.1.02	Execute If Every Flag	EXE	A,T,K, P,R,X, Y	-SCM 1234 OLIU	1:4				Accum. Flags (A) = -SCM Test Flags (T) = OLIU OCK Flags (K) = 1234 Reader Flags (R) = 1234 Punch Flags (P) = 1234 X Flags (X) = 1234 Y Flags (Y) = 1234
9.2.01	Skip If Digit less than Constant	SKL	0:15	0:15	1:4				A = Digit Position B = Digit Value C = No. of Instructions
9.2.02	Execute If Digit less than Constant	EXL	0:15	0:15	1:4				(2) Memory location; May have + or - Relative Addressing
9.3.01	Skip If Accumulator Zero	SKZ	1:4						
9.3.02	Execute If Accumulator Zero	EXZ	1:4						
9.4.01	Compare Alpha	CPA							Label (2)**

** Word = Accumulator: Execute next instruction
Word < Accumulator: Skip next instruction, execute 2nd instruction.
Word > Accumulator: Skip next 2 instructions, execute 3rd instruction.

3.2.10 Miscellaneous

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.			PARAMETERS						
			NUMERIC			SYMBOLIC			
		INSTR.	A	B	C	A	B	C	REMARKS
10.2	No Operation	NOP							
10.1.01	Alarm	ALARM							Sound Alarm once
10.3	Stop	STOP							Return to "Ready" Mode

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
11.6.01	Transfer Receive Buffer	TRB				Label			A = Track number
11.5.01	Load Receive Buffer Register	LRBR				Label (6)			A = Word number
11.5.02	Set Receive Character Pointer	RCP	1:255			Label (1)			A = Character position in buffer
11.5.03	Increment Receive Character Pointer	IRCP	0:255			Label (1)			A = Number of characters
11.6.02	Transfer to Accumulator as Numeric	TRBA	0:16			Label			
11.6.03	Transfer Alpha	TRF	0:255			Label (1)			
11.6.04	Print Alpha from Receive Buffer	PAB	0:150			Label (2)			
11.7.01	Transfer Send Record Area	TSB				Label			A = Track number
11.5.04	Load Keyboard Base Register	LKBR				Label (6)			A = Word number
11.5.05	Set Send Character Pointer	SCP	1:255			Label (1)			A = Character position in buffer
11.7.02	Transfer Accumulator to Send Record Area	TRAB	0:15			Label			A = Number of digits
11.7.04	Transfer Character	TRCB	0:15	0:15		Label			A = USASCII Column number B = USASCII Row No.
11.7.05	Type to Memory	TKM	0:150			Label (1)			A = Number of characters
11.8.01	Retrieve Send Address	RSA							
11.8.02	Retrieve Receive Address	RRA							
11.8.03	Load Send Address Register	LSA							

(1) May have + or - increment value.

(2) Memory location. May have + or - Relative Addressing.

(6) To reference Data Communications Buffers, use no Label; otherwise, must use Label to reference Main Memory Record areas.

3.2.11 Data Communications Instructions (continued)

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		PARAMETERS							
			NUMERIC			SYMBOLIC			
		INSTR.	A	B	C	A	B	C	REMARKS
11.8.04	Load Receive Address Register	LRA							
11.8.05	Retrieve Expected Transmission Number	RTN							
11.8.06	Retrieve Header Transmission Number	RTH							
11.8.07	Load Expected Transmission Number Register	LTN							
11.8.08	Retrieve Send Transmission Number	RSN							
11.8.09	Load Send Transmission Number Register	LSN							
11.8.10	Retrieve Character Pointer Register	RPR							
11.8.11	Load Character Pointer Register	LPR							
11.8.12	Power Off	OFF							
11.8.13	Retrieve Polled Flags	RPF							
11.8.14	Load Polled Flags Register	LPF							
11.9	Skip If Any Flags	SK	R,B,D	1234	1:4				A = Flag group B = Flag number C = No. of instructions
11.9	Execute If Any Flags	EX	R,B,D	1234	1:4				
11.9	Skip If Every Flag	SKE	R,B,D	1234	1:4				
11.9	Execute If Every Flag	EXE	R,B,D	1234	1:4				

3.2.11 Data Communications Instructions (continued)

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
11.9.01	Load Flags	LOD	R	23					A = Flag group B = Flag number
11.9.01	Set Flags	SET	R	23					
11.9.01	Reset Flags	RST	R	23					
11.9.01	Change Flags	CHG	R	23					

R = Reader Flag Group
2 = Message Received Flag
3 = Transmit Ready Flag

B = Buffer Flag Group
2 = Keyboard Buffer Filled Flag
3 = Keyboard Buffer Empty Flag

D = Data Communication Processor Flag Register
1 = Transmission Failure Flag
2 = Message Received Flag
3 = Transmit Ready Flag

3.2.12 Paper Tape Input

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
12.1.01	Read Alpha and Print	RTK	0:150			Label (1)			(1) May have + or - increment value
12.1.02	Read Alpha, Print & Punch	RXTK	0:150			Label (1)			
12.1.03	Read Alpha into Memory & Print	RTKM	0:150			Label (1)			A = No. of characters
12.1.04	Read Alpha into Memory, Print & Punch	RXTKM	0:150			Label (1)			
12.1.05	Read Alpha into Memory, Non-Print	REAM	0:150			Label (1)			
12.1.06	Read Alpha into Memory & Punch, Non-Print	RXEAM	0:150			Label (1)			
12.1.08	Read Numeric into Accumulator	RNK	0:15	0:15		Label			A = digits left of dec. pt. B = digits right of dec. pt.
12.1.09	Release Media Clamp	REL							Open Media Clamp

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part II Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
13.1.01	Type, Punch and Print	XTK	0:150			Label (1)			A = No. of Characters (1) May have + or - increment value
13.1.02	Type to Memory, Punch and Print	XTKM	0:150			Label (1)			
13.1.03	Enter Alpha into Memory & Punch, Non-Print	XEAM	0:150			Label (1)			
13.1.04	Print Alpha and Punch	XPA				Label (2)			
13.1.05	Punch Alpha from Memory, Non-Print	XA				Label (2)			
13.1.06	Punch Code	XC	0:15	0:15		Label			
13.2.01	Print and Punch Numeric	XPN	0:14	0:15		Label			A = Pointer B = Print Mask Address Modifier (2) Memory location; May have + or - Relative Addressing
13.2.02	Print and Punch Numeric, Shift Ribbon if minus	XPNS-	0:14	0:15		Label			
13.2.03	Print and Punch Numeric, Shift Ribbon if plus	XPNS+	0:14	0:15		Label			
13.2.04	Punch Numeric, Non-Print	XN	0:14	0:15		Label			A = Pointer B = Print Mask Address Modifier
13.3.01	Load Punch Count Register	LXC	0:255			Label (1)			
13.3.02	Modify By Punch Count Register	XMOD							
13.3.03	Punch Feed Codes	XB	0:255			Label (1)			

3.2.14 80 Column Card Input

NOTE: To be Published Later.

GP 300 SERIES FIRMWARE INSTRUCTIONS

		PARAMETERS						
		NUMERIC			SYMBOLIC			
	INSTR.	A	B	C	A	B	C	REMARKS
Type and Punch	XTK	0:150			Label (1)			A = No. of characters; follow with SKP
Type Into Memory, Punch and Print	XTKM	0:150			Label (1)*			
Type Into Memory and Punch, Non-print	XEAM	0:150			Label (1)*			
Print and Punch Alpha from Memory	XPA				Label (2)			A = Beginning word; follow with SKP
Punch Alpha from Memory, Non-print	XA				Label (2)			
Punch Code	XC	0:15	0:15					
Print and Punch Numeric	XPN	0:14	0:15					A = Pointer B = Print mask; precede with LPNR
Print and Punch Numeric, Shift Ribbon if Minus	XPNS-	0:14	0:15					
Print and Punch Numeric, Shift Ribbon if Plus	XPNS+	0:14	0:15					
Punch Numeric, Non-print	XN	0:14	0:15					
Load Punch Count Register	LXC	1						
Skip to Column	SKP	1:80						A = Card Column
Duplicate to Column	DUP	1:80						

(1) May have + or - increment value

(2) Word location; may have + or - relative addressing

3.2.16 Pseudo Instructions

GP 300 SERIES FIRMWARE INSTRUCTIONS

Part III Paragraph No.		INSTR.	PARAMETERS NUMERIC A	NUMERIC B	NUMERIC C	SYMBOLIC A	SYMBOLIC B	SYMBOLIC C	REMARKS
3.1.01	Origin	ORG	0: N						
3.1.02	Region	REG	1:255						
3.1.03	Note	NOTE							
3.1.04	Page	PAGE							
3.1.05	Word	WORD							
3.1.06	Advance Line	ADVL	1:4						
3.1.07	Numeric Constant	NUM							
3.1.08	Alphanumeric Constant	ALF							
3.1.09	Print Mask	MASK							
3.1.10	Define	DEF	0:N						
3.1.10	Define	DEFT	0:15	0:15					
3.1.11	Equate	EQU				Label			
3.1.12	End	END							
3.1.13	Library Routine*	LIB				Label (5)			(5) May have label up to 10 characters
3.1.14	Documentation*	DOC							
3.1.15	Establish Buffer	ESTB							
3.1.16	Code	CODE	(7)						(7) A = 4 hexadecimal digits

* For use only on B 3500/B 5500 versions of TC 500 Assembler

APPENDIX A-1

TC 500 MAIN MEMORY FIRMWARE FOR GP 300 PROGRAMING LANGUAGE

TO BE PUBLISHED LATER.

APPENDIX A-2

FIRMWARE SET "i" INSTRUCTION LIST (GP 300 INSTRUCTIONS)

(Instructions listed in boxes are NOT included in list i)

NUMERIC KEYBOARD

NK
NKR
NKCM
NKRCM

ALPHA KEYBOARD

TK
TKM
LKBR
PKA 1234
LPKR

EAM
PKB

PRINT

PN
PC
PCP
PA
POS
LPNR
RR

PNS-
PNS+
PC-
PC+

FORMS

CC
LLCR
LRCR
LLLR
LRLR
AL
AR
ALTO
ARTO

OC
CC
ALR

ARITHMETIC

ADA
SUA
TRA
TRM
CLA
INK
ADK
SUK

ADM
SUM
CLM
MUL
MULR
DIV
REM
SLRO
SLROS
LSR

FLAGS

LOD
SET
RST
CHG

INDEX REGISTER (1)

LIR
IIR
MOD

ADIR
DIR
TAIR

BRANCH

BRU

SRJ
SRR

SKIP/EXECUTE

SK
SKE
EX
EXE
SKL
EXL
SKZ
EXZ

CPA

MISCELLANEOUS

NOP
ALARM
STOP

DATA COMMUNICATIONS

TRB
LRBR
RCP
TRF
PAB
TSB
LKBR
SCP
TRAB
TRCB
TKM
RPR
LPR
RPF
LPF

IRCP
TRBA
RSA
RRA
LSA
LRA
RTN
LTN
RSN
LSN
RTH
OFF

Appendix B

TC 500 CHARACTER SETS

The USASCII and Commercial character sets for the TC 500 are listed below in their collating sequence in ascending order. Each character set consists of 64 graphic characters, the Space code, and the End of Alpha code. The USASCII character set consists of the USASCII characters in columns 2, 3, 4, and 5 of the USASCII table, plus End of Alpha (NUL) and Overline. Those Commercial characters that differ from the USASCII characters are shown in parentheses.

The internal or machine language code for each character is given; this code consists of two hexadecimal digits which correspond to the column and row number of the character in the USASCII table (A = row 10, B = 11, C = 12, D = 13, E = 14, F = 15). In addition, the decimal value of each character is given as required when using Index Registers for modification.

Character	Internal Code		Indexing Value	Character	Internal Code		Indexing Value	Character	Internal Code		Indexing Value	Character	Internal Code		Indexing Value
End of Alpha (NUL)	0	0	0												
Space	2	0	32	0	3	0	48	@	4	0	64	P	5	0	80
!	2	1	33	1	3	1	49	A	4	1	65	Q	5	1	81
”	2	2	34	2	3	2	50	B	4	2	66	R	5	2	82
#	2	3	35	3	3	3	51	C	4	3	67	S	5	3	83
$	2	4	36	4	3	4	52	D	4	4	68	T	5	4	84
%	2	5	37	5	3	5	53	E	4	5	69	U	5	5	85
&	2	6	38	6	3	6	54	F	4	6	70	V	5	6	86
’	2	7	39	7	3	7	55	G	4	7	71	W	5	7	87
(	2	8	40	8	3	8	56	H	4	8	72	X	5	8	88
)	2	9	41	9	3	9	57	I	4	9	73	Y	5	9	89
*	2	A	42	:	3	A	58	J	4	A	74	Z	5	A	90
+	2	B	43	;	3	B	59	K	4	B	75	[(¾)	5	B	91
,	2	C	44	< (½)	3	C	60	L	4	C	76	\ (¢)	5	C	92
-	2	D	45	=	3	D	61	M	4	D	77	](CR)	5	D	93
.	2	E	46	> (¼)	3	E	62	N	4	E	78	^ (°)	5	E	94
/	2	F	47	?	3	F	63	O	4	F	79	_	5	F	95
												~ (<>)	7	E	126
												DEL	7	F	127

cut along dotted line

BURROUGHS CORPORATION
TECHNICAL PUBLICATIONS
REMARKS FORM

TITLE: _____ FORM: _____

_____ DATE: _____

_____

CHECK TYPE OF SUGGESTION:

☐ ADDITION ☐ DELETION ☐ REVISION ☐ ERROR

GENERAL COMMENTS AND/OR SUGGESTIONS FOR IMPROVEMENT OF PUBLICATION:

FROM: NAME _____ DATE _____

TITLE _____

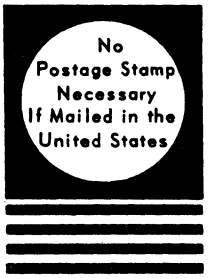
COMPANY _____

ADDRESS _____

_____

STAPLE

FOLD DOWN SECOND FOLD DOWN



BUSINESS REPLY MAIL
First Class Permit No. 817, Detroit, Mich. 48232

Burroughs Corporation
6071 Second Avenue
Detroit, Michigan 48232

attn: Sales Technical Services
Group II

FOLD UP FIRST FOLD UP