#### TELECOMMUNICATIONS INPUT OR MULTICHARACTER INPUT WITH MULTI-SPECIAL-CHARACTER FEATURE

( A sequence inside [ ] is optional, execution depends upon action atom for the incoming character)

Code	and the same of the		Signal Ser	quenc	in the second		
FOh, h,	(CPB, IBS,	[WR,	ECHO1/OBS]	[WR,	ECHO2/OBS].	SAVE DATA	, REPEAT
F1h,h	(CPB, IBS,	1	ECHO1/OBS],	1	ECHO2/OBS], [	SAVE DATA	REPEAT
F4h,h	(CPB, IBS,	[WR,	ECHO1/CBS],	[WR,	ECHO2/CBS], [	SAVE DATA!	REPEAT
F5h,h	(CPB, IBS,	1	ECHO1/CBS),	1	ECHOZ/CBS], [	SAVE DATA	REPEAT
F8h,h	(CPB, IBS, MASK,	[WR,	ECHO1/OBS],	[WR	ECHO2/OBS],	SAVE DATA	, REPEAT
F9h,h	(CPB, IBS, MASK,	1	ECHO1/OBS),	1	ECHO2/OBS], [	SAVE DATA	REPEAT
FCh, h,	(CPB, IBS, MASK,	[WR.	ECHO1/CBS),	[WR,	ECHO2/CBS], [	SAVE DATA!	REPEAT
FDh,h,	(CPB, IBS, MASK,	1	ECHO1/CBS).	1	ECHO2/CBS], [	SAVE DATALL	REPEAT

Action atom for any incoming character not matching a special character.

## LEGEND (for Fh<sub>2</sub>h<sub>3</sub>h<sub>4</sub> microcommands only)

Mnemonic	Operation
СРВ	CPU sets Ready/Busy signal level to Ready.
ECHO1/CBS	CPU sends echo of received character with CBS strobe to CO device.
ECHO1/OBS	CPU sends echo of received character with OBS strobe to CO device.
ECHO2/CBS	CPU sends echo of received character with CBS strobe to output channel of input device.*
ECHO2/OBS	CPU sends echo of received character with OBS strobe to output channel of input device.*
IBS	CPU awaits input strobe from enabled device.**
MASK	Set high-order eighth bit of received character to zero.
REPEAT	Repeat sequence in parentheses until valid termination condition detected.
SAVE DATA	Save received character in data buffer .
WR	CPU awaits Ready signal from enabled device.
	*An Fh <sub>2</sub> h <sub>3</sub> h <sub>4</sub> microcommand ignores any preceding address strobe of the form 7h <sub>2</sub> h <sub>3</sub> h <sub>4</sub> and uses
	the address specified by the \$GIO statement.

#### REGISTER USAGE (for \$GIO statements having an Fh 2h3h4 microcommand)

Register (Byte)	Bit Position	Use
1	all	Automatic storage of h <sub>3</sub> h <sub>4</sub> , specified in the microcommand (with h <sub>4</sub> set to 0). The stored value is the action atom for any input character not matching a character in the special character list.
2, 3, 4, 5	all	Not used.
6	all	Automatic storage of character received with ENDI-level = 1,
7	all	Not used,
8	01	1 = Buffer overflow.
	02	1 = Element overflow,
	04	Not used.
	08	Not used.
	10	1 = Timeout.
	20	1 = ENDI-level termination.
	40	1 = Terminator-character termination.
	80	1 = Separator received for last element.
9, 10	all	Automatic storage of the count of elements used for incoming data
11,		Storage of special character list (atom, character, atom, character, etc.). The list must end with HEX (2020).



<sup>\*\*</sup>A separator denotes the end of an input "line"; the next received character is stored as the first character in the next element of the \$GIO buffer. (If a separator is received for the last element, the microcommand is terminated.)

#### THE SGIO STATEMENT



Example: 100 \$GIO WRITE /03B (6C01 4400 A206 8601, R\$) B\$() <5,90>

- The microcommand sequence must be one or more groups of four hexdigits (h, h, h, h, h) representing
  valid codes from the microcommand categories recognized by the System 2200. The microcommand
  sequence can be specified directly, as shown in the example, or indirectly by an alphanumeric variable into which the appropriate hexdigit values have been previously stored.
- The error/status/general-purpose registers must be represented by an alphanumeric variable at least
  10 bytes long (12 or more bytes are needed if an Fh<sub>2</sub>h<sub>3</sub>h<sub>4</sub> microcommand is used). The byte-positions
  in the alphanumeric variable are called "registers" to emphasize the multi-purpose usage of the variable.
   The data buffer is needed only if the microcommand sequence includes a multicharacter data transfer
- The oas outer a messed only it the microcomman sequence includes a midmetaracter data trainer microcomman of the form  $Ah, h_h$ ,  $h_h$ ,  $h_h$ ,  $h_h$ ,  $h_h$ ,  $h_h$ ,  $h_h$ . The SGIO buffer can be represented by an alpha variable, a string (STR) function, an alpha array designator (i.e., an alpha array designator having a field expression specifying the portion of the array to be used for data output or input). The field expression format is as follows:

<s, n> for any \$GIO statement not having an Fh2h3h4 microcommand <s, m, e> for a \$GIO statement having an Fh2 h3 h4 microcommand

- n = number of consecutive bytes m = number of bytes per element e = number of elements.

Code	Operation	
Oh <sub>2</sub> h <sub>3</sub> h <sub>4</sub> , 1h <sub>2</sub> h <sub>3</sub> h <sub>4</sub>	Control	
4h <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 0 through 7)	Single character output	
5h <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 0 through 7)	Single character output with acknowledge	
6h <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>3</sub> = 0 through F)	Single character output with echo	no data
7h <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 1 or 3)	Single address strobe	buffer
8h <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 0, 2, 8, A)	Single character input with verify	required
86h <sub>3</sub> h <sub>4</sub>	Single character input	
9h <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 2, 3, 6, 7)	Single character input with echo	
Ah <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 0, 1, 2, 4, 5, 6)	Multicharacter output	
Bh <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 0, 1, 4, 5,)	Multicharacter output with acknowledge	
Bh <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 2, 3, 6, 7)	Multicharacter output with echo	
Bh <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 8, 9, C, D)	Multicharacter output (each character requested)	data
BAh <sub>3</sub> h <sub>4</sub>	Multicharacter verify	buffer required
Ch <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 2, 6)	Multicharacter input	radamen
Ch <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 0, 1, 4, 5)	Multicharacter input with echo	
Ch <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 8 through F)	Multicharacter input (each character requested)	
Fh <sub>2</sub> h <sub>3</sub> h <sub>4</sub> (h <sub>2</sub> = 0, 1, 4, 5, 8, 9, 0	C, D) Telecommunications input	

LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876 TEL (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

Printed in U.S.A. 700-3782 8-76-10M

## **SYSTEM 2200**



# **MICROCOMMANDS**

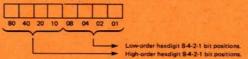


<sup>\*\*\*</sup> A terminator denotes the end of a data stream; the microcommand is terminated,

## REGISTER USAGE (for \$GIO statements not having an Fh<sub>2</sub>h<sub>3</sub>h<sub>4</sub> microcommand)

Register (Byte)	Bit Position*	Use			
1	all	General-purpose or storage of a special termination character.			
2, 3, 4	all	General-purpose.			
5	all	General-purpose or automatic storage of an LRC character.			
6	alt	General-purpose or automatic storage of an ENDI-level=1 character.			
7.	all	General-purpose.			
8	01	1 = Buffer overflow.			
	02	1 = LRC error.			
	04	1 = Echo/Verify error.			
	08	1 = Compare error.			
	10	1 = Timeout.			
	20	1 = ENDI-level termination.			
	40	1 = Special character termination.			
	80	1 = Count termination,			
9, 10	all	Automatic storage of the count of transferred characters for a multicharacter output or input microcommand.			

## \*Bit position labels for status code (register 8) are as follows:



#### LEGEND (for all microcommands except FH\_h\_h\_h\_)

Mnemonic	Operation
ABS	CPU sends an "address bus strobe" with an immediate or indirect address to deselect the current address and select the specified address.
CBS	CPU sends a CBS strobe to the enabled device.
CHECK T. T1, T2	CPU checks for the termination condition specified by ha.
CPB	CPU sets its Ready/Busy signal to Ready.
DATAOUT	CPU sends out next character from \$GIO buffer.
ECHO	CPU sends echo of received character to enabled device.
IBS	CPU awaits input strobe from enabled device.
IMM	Immediate character is HEX(h <sub>1</sub> h <sub>4</sub> ), specified by the microcommand.
IND	Indirect character is in the register specified by h3.
LEND	CPU executes the LRC end sequence specified by h <sub>4</sub> .
OBS	CPU sends an OBS strobe to the enabled device.
REPEAT	CPU repeats the sequence in parentheses for each character in a multicharacter input or output operation.
SAVE	CPU saves received character in the register specified by h <sub>4</sub> .
SAVE DATA	CPU saves received character in the \$GIO buffer.
SAVE LRC	CPU saves calculated LRC character in register 5.
SEND LRC	CPU sends calculated LRC character to enabled device.
TERM	CPU terminates \$GIO statement if compared characters are not equal.
VERIFY	CPU compares received character; if unequal, the echo/verify error bit (bit 04 in register 8) is set to 1.
WR	CPU awaits Ready signal from enabled device.
W5	CPU waits 5 microseconds.

#### CONTROL MICROCOMMANDS

Code	Operation
Oh <sub>2</sub> h <sub>3</sub> h <sub>4</sub>	Store character HEX (h <sub>3</sub> h <sub>4</sub> ) in register h <sub>2</sub> .
11h <sub>3</sub> h <sub>4</sub>	Copy contents of register h <sub>3</sub> to register h <sub>4</sub> .
1200	Disable previously set delay/timeout condition,
12h <sub>3</sub> 1	Introduce a delay* before each subsequent output strobe (except ABS); the interval in units of 50 microseconds is specified in binary in registers $h_3$ and $h_3+1$ , where $1\!\leq\!h_3\!\leq\!6$ . Maximum delay interval = HEX (FFFF) $\sim3.3$ seconds.
12h <sub>3</sub> 2	Initiate a timeout* before sensing each subsequent ready signal or input strobe; the interval in units of 1 millisecond is specified in binary in registers $h_3$ and $h_3+1$ , where $1\leq h_3\leq 6$ . Maximum timeout interval = HEX (FFFF) $\sim 65.5$ seconds. If a timeout interval is exceeded, set error bit (bit 10 in register 8) and terminate.
14h <sub>3</sub> h <sub>4</sub>	If contents of register $h_3 \neq$ contents of register $h_4$ , set compare error bit (bit 08, register 8) to 1.
15h <sub>3</sub> h <sub>4</sub>	If contents of register h <sub>3</sub> ≠ contents of register h <sub>4</sub> , set compare error bit (bit 08, register 8) and terminate.
16h <sub>3</sub> h <sub>4</sub>	If complemented status (register 8) code AND $h_3h_4 \neq HEX(00)$ , terminate (i.e., terminate if any bit specified by the mask $h_3h_4$ is equal to zero).
17h <sub>3</sub> h <sub>4</sub>	If status (register 8) code AND $h_3h_4 \neq HEX(00)$ , terminate (i.e., terminate if any bit specified by the mask $h_3h_4$ is equal to one).
	*Delay and timeout conditions are mutually exclusive. Also, neither a delay nor a timeout can be in effect during execution of a multicharacter output with seb microcommand of the form 82h <sub>3</sub> h <sub>3</sub> . 8h, 3h, ye of 3h, ye if 3h, or 8 shb, ye if so, a fall indication of an exbo error may occur.

## SINGLE CHARACTER OUTPUT AND ADDRESS STROBE

	The second secon	To be sent	To be saved
200100	Single Character Output		
40h, h.	WR, OBS/IMM	HEX (h,h,)	1000
41h,h,	OBS/IMM	HEX (h <sub>3</sub> h <sub>4</sub> )	3450
42h,0	WR, OBS/IND	from register h,	
43h,0	OBS/IND	from register ha	
44h,h,	WR, CBS/IMM	HEX (h,h,)	100
45h,h,	CBS/IMM	HEX (h <sub>2</sub> h <sub>4</sub> )	
46h, 0	WR, CBS/IND	from register h,	
47h,0	CBS/IND	from register h <sub>3</sub>	
50h, h,	Single Character Output with Acknowledge WR, OBS/IMM, W5, CPB, IBS	HEX (h <sub>3</sub> h <sub>4</sub> )	
51h,h,	OBS/IMM, W5, CPB, IBS	HEX (h <sub>3</sub> h <sub>4</sub> )	1.72
52h, h.	WR, OBS/IND, W5, CPB, IBS, SAVE	from register h,	in register h.
53h, h,	OBS/IND, W5, CPB, IBS, SAVE	from register h.	in register h.
54h, h,	WR, CBS/IMM, W5, CPB, IBS	HEX (h <sub>2</sub> h <sub>4</sub> )	Marie Control
55h, h,	CBS/IMM, W5, CPB, IBS	HEX (h <sub>2</sub> h <sub>4</sub> )	
56h,h,	WR, CBS/IND, W5, CPB, IBS, SAVE		in register ha
57h, h.	CBS/IND, W5, CPB, IBS, SAVE	from register h <sub>3</sub>	in register h.
	Single Character Output with Echo		11 11 11
60h, h,	WR, OBS/IMM, W5, CPB, IBS, VERIFY	HEX (h <sub>3</sub> h <sub>4</sub> )	
61h, h,	OBS/IMM, W5, CPB, IBS, VERIFY	HEX (h <sub>3</sub> h <sub>4</sub> )	
62h, h,	WR, OBS/IND, W6, CPB, IBS, SAVE, VERIFY	from register h <sub>3</sub>	
63h, h,	OBS/IND, W6, CPB, IBS, SAVE, VERIFY	from register h;	in register h <sub>4</sub>
64h, h,	WR, CBS/IMM, W5, CPB, IBS, VERIFY	HEX (h <sub>3</sub> h <sub>4</sub> )	
65h, h <sub>4</sub>	CBS/IMM, W5, CPB, IBS, VERIFY	HEX (h <sub>3</sub> h <sub>4</sub> )	
66h, h,	WR, CBS/IND, W5, CPB, IBS, SAVE, VERIFY	from register h <sub>3</sub>	in register h <sub>4</sub>
67h, h,	CBS/IND, W6, CPB, IBS, SAVE, VERIFY	from register ha	in register ha
68h, h,	WR, OBS/IMM, W5, CPB, IBS, VERIFY, TERM		
69h,h,	OBS/IMM, W5, CPB, IBS, VERIFY, TERM		
6Ah,h	WR, OBS/IND, W5, CPB, IBS, SAVE, VERIFY, TERM	from register h <sub>3</sub>	
6Bh,h,	OBS/IND, W5, CPB, IBS, SAVE, VERIFY, TERM	from register h <sub>3</sub>	in register he
6Ch <sub>3</sub> h <sub>4</sub>	WR, CBS/IMM, W5, CPB, IBS, VERIFY, TERM	HEX (h <sub>3</sub> h <sub>4</sub> )	
6Dh <sub>3</sub> h <sub>4</sub>	CBS/IMM, W5, CPB, IBS, VERIFY, TERM		
6Eh,h,	WR, CBS/IND, WS, CPB, IBS, SAVE, VERIFY, TERM	from register h <sub>3</sub>	111 5 15 15 15 15 15 15
6Fh <sub>3</sub> h <sub>4</sub>	CBS/IND, W5, CPB, IBS, SAVE, VERIFY, TERM	from register h <sub>3</sub>	in register ha
	Address Strobe	HER IS S. I.	
71h, h, 73h, 0	ABS/IMM ABS/IND	HEX (h <sub>3</sub> h <sub>4</sub> ) from register h <sub>3</sub>	

## SINGLE CHARACTER INPUT

Code	Signal Sequence	Verify Character	Character To Be Saved
	Single Character Input		TO SOLL !
8600 860h <sub>4</sub>	CPB, IBS, CPB, IBS, SAVE,		in register h <sub>4</sub>
SPER	Single Character Input with Verify	200	11111111
80h <sub>3</sub> h <sub>4</sub>	CPB, IBS, VERIFY/IMM	HEX(h <sub>3</sub> h <sub>4</sub> )	
82h3h4	CPB, IBS, SAVE, VERIFY/IND	in register ha	in register h <sub>4</sub>
88h <sub>3</sub> h <sub>4</sub>	CPB, IBS, VERIFY/IMM, TERM	HEX(h <sub>3</sub> h <sub>4</sub> )	
8Ah <sub>3</sub> h <sub>4</sub>	CPB, IBS, SAVE, VERIFY/IND, TERM	in register h <sub>3</sub>	in register h <sub>4</sub>
1	Single Character Input with Echo		04 377
9200	CPB, IBS, WR, ECHO/OBS	1000	
920h <sub>4</sub>	CPB, IBS, SAVE, WR, ECHO/OBS	100	in register ha
9300	CPB, IBS, ECHO/OBS		
930h <sub>4</sub>	CPB, IBS, SAVE, ECHO/OBS		in register h <sub>4</sub>
9600	CPB, IBS, WR, ECHO/CBS		
960h4	CPB, IBS, SAVE, WR, ECHO/CBS	1 1 0	in register h <sub>4</sub>
9700	CPB, IBS, ECHO/CBS		
970h4	CPB, IBS, SAVE, ECHO/CBS		in register ha

MULTICHARACTER OUTPUT

(A sequence in parentheses is repeated for each character in the data buffer)

Code	Signal Sequence			Lenc
	Multicharacter Output		1	
A00h	(WR, DATAOUT/OBS), REPEAT, LEND			h,
A10h,	( DATAOUT/OBS), REPEAT, LEND		50	b,
A20h	high speed version of A00h <sub>4</sub> ; no timeout or del	ay .		h,
A40h	(WR, DATAOUT/CBS), REPEAT, LEND		32	h,
A50h	( DATAOUT/CBS), REPEAT, LEND			h,
A60h	SCAN DATA BUFFER, CALCULATE LRC, LI	END	100	h <sub>4</sub>
	Multicharacter Output with Acknowledge		100	
BOh,h,	(WR, DATAOUT/OBS, W5, CPB, IBS,	CHECK T), REPEAT, LEND	h,	h,
B1h,h,	( DATAOUT/OBS, W5, CPB, IBS,	CHECK T), REPEAT, LEND	h,	h.
84h, h,	(WR, DATAOUT/CBS, W5, CPB, IBS,	CHECK T), REPEAT, LEND	h,	h,
85h,h,	( DATAOUT/CBS, W5, CPB, IBS,	CHECK T), REPEAT, LEND	h <sub>3</sub>	b <sub>a</sub>
	Multicharacter Output with Echo	Saula Made	11.	
82h, h.	(WR. DATAOUT/OBS, W5, CPB, IBS, VERIFY	CHECK T), REPEAT, LEND	h,	h,
B3h, h,	( DATAOUT/OBS, W5, CPB, IBS, VERIFY	CHECK T), REPEAT, LEND	h <sub>3</sub>	· he
86h, h,	(WR, DATAOUT/CBS, W5, CPB, IBS, VERIFY	CHECK TI, REPEAT, LEND	h,	h <sub>4</sub>
87h,h,	( DATAOUT/CBS, W5, CPB, IBS, VERIFY	CHECK TI, REPEAT, LEND	h <sub>3</sub>	h <sub>a</sub>
And the second	Multicharacter Output with each Characte		100	
B8h, h,	(CPB, IBS, CHECK T, WR, DATAQUT/OBS), I		h <sub>3</sub>	h.
89h, h,	(CPB, IBS, CHECK T, DATAOUT/OBS), F	Control of the Contro	h,	h <sub>e</sub>
BCh,h4	(CPB, IBS, CHECK T, WR, DATAOUT/CBS), F		h <sub>i</sub>	h <sub>a</sub>
BDh,h,	(CPB, IBS, CHECK T. DATAOUT/CBS), F	REPEAT, LEND	h <sub>3</sub>	h <sub>a</sub>
BAh,0	Multicharacter Verify (CPB, IBS, VERIFY, CHECK T), REPEAT		h <sub>3</sub> .	-1

ormination Microcommend Condition	80. I	8. B	2. 86. 3. 87.	88. BC 88. BD	BA.0	
None Igo to next microcommend). Terminate if verify unequal. Yerminate if ENDI lavel -logic "t". Terminate on either condition.	0		0 1 2 3	2	0) 40	
Microcommand LRC End Sequence	A.	BG. three		. BO.		
None (po to neat microcommend) MR, SEND LRC/OBS, SAVE LRC SEND LRC/OBS, SAVE LRC MR, SEND LRC/CBS, SAVE LRC SEND LRC/CBS, SAVE LRC	0 2 3 4 6	0 2 3 4 6	T	4	4	*

#### MULTICHARACTER INPUT

(A sequence in parentheses is repeated until a valid termination condition occurs)

Code	Signal Sequence	Check T	Lend
	Multicharacter Input		
CZZNA	(CPB, IBS, no timeout or delay, CHECK ENDI, SAVE DATA), REPEAT, LEND	2	Pta
CSh <sub>3</sub> h <sub>4</sub>	(CPB, IBS, ECHO/CBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>3</sub>	ha
D.	Multicharacter Input with Echo		
COh <sub>3</sub> h <sub>4</sub>	(CPB, IBS, WR, ECHO/OBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	No.	Pla.
Cihsha	(CPB, IBS, ECHO/OBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	hi	Re
C4h <sub>3</sub> h <sub>4</sub>	(CPB, IBS, WR, ECHO/CBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>3</sub>	h <sub>4</sub>
C5h <sub>3</sub> h <sub>4</sub>	(CPB, IBS, ECHO/CBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>3</sub>	h <sub>4</sub>
	Multicharacter Input with Each Character Requested		
C8h <sub>3</sub> h <sub>4</sub>	(WR, OBS, WB, CPB, IBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>3</sub>	n <sub>4</sub>
CShaha	L OBS, W5, CP8, IBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>3</sub>	n.
CAh <sub>3</sub> h <sub>4</sub>	ICPB, WR, OBS, IBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>2</sub>	ha
CBh <sub>1</sub> h <sub>4</sub>	(CPB. OBS, IBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>3</sub>	h.
CCh <sub>3</sub> h <sub>4</sub>	(WR, CBS, W6, CPB, IBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>3</sub>	h.
COh <sub>a</sub> h <sub>a</sub>	( CBS, W5, CPB, IBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	No.	Na.
CEh <sub>2</sub> h <sub>4</sub>	(CPB, WR, CBS, IBS, CHECK TI, SAVE DATA, CHECK T2), REPEAT, LEND	h <sub>2</sub>	- he
CFh <sub>3</sub> h <sub>4</sub>	(CPB. CBS, IBS, CHECK T1, SAVE DATA, CHECK T2), REPEAT, LEND	No.	No.

Variation Truster Trus check (do not seve char.) 2 check (save cher, in reg. 8) 3 check (seve cher, in reg. 8) sheck (do not seve char.) 6 chack (sere char, in reg. 6) 7 check (seve char, in reg. 6) check (do not seve char.)

	Valid "Lend" Codes
N	LRC End Sequence
0	None Igo to next microcommend).
1	Calculate LRC and save.
2	Calculate LRC, save, compare with ENOt character, and set LRC error bit luse only if hy = 21.