



Keyboard layout:

- CPX may be combined with all stack operations and +, -, x, /, ±, x<sup>2</sup>, √x, 1/x, //, Γ, |x|, RND, as well as (HYP) SIN, COS, TAN, LN, LG<sub>y</sub> and their inverses
- Modes are HMS, 2, 8, 10, 16, .d, b/c, FIX, SCI, ENG, DEG, RAD, GRAD
- → may be combined with H, HMS, DEG, RAD



Active operations in hexadecimal mode.  $\rightarrow$  is for addressing only (see below). The primary functions of the top left 6 keys are numeric input, so the default primary functions of these keys are accessed using f-shift. – In the other integer modes, the active keyboard will look alike, but the default primary functions of the top left 6 keys will stay as they are in DECM.



Active operations in alpha mode.  $\rightarrow$  is for addressing only (see below). The primary function of most keys will be inserting a letter. The basic arithmetic keys,  $\Sigma$ ,  $\pi$ , and “!” will do so, too. To reach the default primary functions, f-shift will be necessary wherever a letter stands next to a key.

## ADDRESSING REGISTERS

1	User input	$x=?$ or one of the other comparisons			<b>RCL</b> , <b>STO</b> , <b>VIEW</b> , $x\geq$ , <b>SF</b> , <b>CF</b> , or <b>FS?</b>	
	Display	<b>OP _</b> (e.g. $x > \_$ ) Alpha mode is set.			<b>OP _</b> (e.g. <b>RCL</b> $\_$ )	
2	User input <sup>1</sup>	<b>0</b>	<b>Y</b> , <b>Z</b> , <b>T</b> , or <b>L</b>	<b>ENTER</b> ↑	<b>ENTER</b> ↑	Register or flag number <sup>2</sup>
	Display	<b>OP 0</b> e.g. $x \leq 0$	<b>OP x</b> e.g. $x \geq y$	<b>OP r _</b>	<b>OP s _</b> Alpha mode is set.	<b>OP nn</b> e.g. $x \langle \rangle 15$
3	User input	Register number <sup>2</sup>			<b>ENTER</b> ↑	Register number <sup>2</sup>
	Display	<b>OP r nn</b> e.g. $x \neq r23$			<b>OP s x</b> e.g. <b>STO</b> $sZ$	<b>OP →s _</b> Alpha mode is set.
4		Store $x$ on stack level $z$ .			<b>X</b> , <b>Y</b> , <b>Z</b> , <b>T</b> , or <b>L</b>	Recall the content of the register where register 3 is pointing to.
	Display	Show the content of the register where <b>LASTx</b> is pointing to.			<b>OP →s x</b> e.g. <b>VIEW</b> $\rightarrow sL$	

<sup>1</sup> For **RCL** and **STO**, an arithmetic operator (+, −, ×, /, ^) may precede step 2.

<sup>2</sup> Register number may be **0 0** ... **9 9**. For any of the lowest 10 registers, you may enter e.g. **5 ENTER**↑ instead of **0 5**.

## ADDRESSING LABELS

1	User input	<b>GTO</b> or <b>XEQ</b> , <b>LBL</b> , <b>SOLVE</b> , <b>INTEG</b> <sup>3</sup>	
	Display	<b>OP “_”</b> (e.g. <b>GTO _</b> ) Alpha mode is set	
2	User input	Label + <b>ENTER</b> <sup>4</sup>	<b>ENTER</b> <b>→</b>
	Display	<b>OP “name”</b> e.g. <b>SLV“STF”</b> Solve the function <b>STF</b> (with STF keyed in).	<b>OP → _</b>
3	User input		<b>ENTER</b>
	Display		<b>OP →s _</b> Alpha mode is set.
			Register number <b>OP → nn</b> e.g. <b>XEQ →03</b>
4	User input		<b>X</b> , <b>Y</b> , <b>Z</b> , <b>T</b> , or <b>L</b>
	Display	Integrate the function which’s label is on stack level <b>y</b> .	<b>OP →s x</b> e.g. <b>INT →sY</b> Execute the routine which’s label is in register <b>3</b> .


<sup>3</sup> **SOLVE** and **INTEG** will be displayed as **SLV** and **INT**, respectively. No indirect addressing with **LBL**.

<sup>4</sup> A label may consist of up to 3 alphanumeric characters. **ENTER** is only needed if less than 3 characters are entered.

## INDICATORS

There are a number of indicators signaling the mode the calculator is running in. The defaults DECM and DEG are not indicated.













Indicator	a	2	d	h	8	PRG
<b>Set by operation</b>	AON X.FCN	BINM	IDECM	HEXM	OCTM	PRGON
<b>Cleared by operation</b>	AOFF	DECM FRACM HEXM IDECM OCTM	BINM DECM FRACM HEXM OCTM	BINM DECM FRACM IDECM OCTM	BINM DECM FRACM HEXM IDECM	PRGOFF

Indicator	GRAD	HMS	RAD	/c
<b>Set by operation</b>	GRAD	HMSM >HMS TIMER	RAD ACOSH ASINH ATANH	FRACM, 2 <sup>nd</sup>  in input ( \HMS)
<b>Cleared by operation</b>	DEG RAD >HR ACOSH ASINH ATANH	DECM >HR	DEG GRAD >HR	BINM DECM IDECM HEXM OCTM

Within integer modes, the annunciator “c” shows a carry bit set.

## INDEX OF OPERATIONS

This lists all functions available on the 34S with the necessary keystrokes. Functions accessible via X.FCN will show up with their name unless specified differently explicitly. Generally, integer functions will work like in 16C, and the other like in 42S. If no parameters are specified though required, then they will be taken from the stack. The modes are abbreviated by their indicators.

Name	Keys to press	Works in modes	Remarks
A ... F	 etc.	h	The top left 6 keys will work for numeric input in hex mode by default.
ABS	 	\a	
ACOS	 	DECM	
ACOSH	  	DECM	
ALL	  ...	DECM	
AND	 	2, 8, d, h	

Name	Keys to press	Works in modes	Remarks
AOFF	ALPHA	a	Toggles alpha mode. *) Please see the table for register addressing above for details.
	ENTER↑		
AON	ALPHA	\a	
	ENTER↑	*)	
ASIN	SIN <sup>-1</sup>	DECM	
ASINH	HYP <sup>-1</sup>	DECM	
ASR	X.FCN ...	2, 8, d, h	
ATAN	TAN <sup>-1</sup>	DECM	
ATANH	HYP <sup>-1</sup>	DECM	
A0	L.R.	DECM	Calculates the intercept of the fit curve according to the fit model chosen.
A1	L.R.	DECM	Calculates the slope of the fit curve according to the fit model chosen.
BASE+		2, 8, d, h	
BASE-		2, 8, d, h	
BASE×		2, 8, d, h	
BASE/		2, 8, d, h	
BASE+/-		2, 8, d, h	
BC?	X.FCN ...	2, 8, d, h	
BINM	2	\a	Sets binary mode.
BST		All	
BS?	X.FCN ...	2, 8, d, h	
CB	X.FCN ...	2, 8, d, h	
CF	CF	All	
CHS		DECM	
CLALL	CLALL	All	

Name	Keys to press	Works in modes	Remarks
CLPR	<b>h</b> <b>CLPR</b>	PRG	
CLRG	<b>h</b> <b>CLRG</b>	All	
CLST	<b>0</b> <b>h</b> <b>FILL</b>	All	
<b>CLX</b>	<b>h</b> <b>CLx</b>	All	<b>CPX</b> <b>CLx</b> clears <b>x</b> and <b>y</b> .
CLΣ	<b>h</b> <b>CLΣ</b>	DECM	
COMB	<b>f</b> <b>Cx.y</b>	DECM	
COMPLEX	<b>CPX</b> ...	DECM	Indicating complex operations, acting on <b>x</b> and <b>y</b> , where <b>x</b> contains the real part and <b>y</b> the imaginary of the complex number. This key may be combined with any function which's name is printed in <i>italics</i> in this table.
CONJ	<b>h</b> <b>CONJ</b>	DECM	Changes the sign of <b>y</b> .
CORR	<b>g</b> <b>r</b>	DECM	
<b>COS</b>	<b>f</b> <b>COS</b>	DECM	
<b>COSH</b>	<b>f</b> <b>HYP</b> <b>COS</b>	DECM	
DECM	<b>f</b> <b>.d</b>	\a	HMS data in <b>x</b> will be converted to decimal.
DEG	<b>h</b> <b>DEG</b>	\a	
DENMAX	<b>h</b> <b>X.FCN</b> /c n	DECM	Sets the maximum denominator for fractions.
DSE	<b>f</b> <b>DSE</b> address	PRG	
EEX	<b>E</b>	DECM	
ENG	<b>h</b> <b>ENG</b> #	DECM	
ENTER↑	<b>ENTER↑</b>	All	
EXPF	<b>h</b> <b>X.FCN</b> ...	DECM	Selects the exponential curve fit model.
<b>E↑X</b>	<b>g</b> <b>e<sup>x</sup></b>	DECM	
<b>E↑X-1</b>	<b>h</b> <b>X.FCN</b> ...	DECM	
FACT	<b>h</b> <b>!</b>	DECM	
FCSTX	<b>h</b> <b>X.FCN</b> ...	DECM	



Name	Keys to press	Works in modes	Remarks
FCSTY		DECM	
FC?	...	All	
FILL		All	Copies <b>x</b> in <b>y</b> , <b>z</b> , and <b>t</b> .
FIX	#	DECM	
FP		DECM	
FRACM		DECM	Sets fraction mode like in HP32SII.
FS?		All	
<b>GAMMA</b>	...	DECM	
GRAD			
GTO	label	PRG	Like in HP32S
		\PRG	
	label	\PRG	
		\PRG	
HEXM		\a	Sets hexadecimal mode.
HMSM		DECM	Sets HMS mode.
HMS+		HMS	
HMS-		HMS	
IDECM		\a	Sets integer decimal mode.
INTEG	label	DECM	Parameters will be transferred like in 15C.
IP		DECM	
ISG		PRG	
<b>LASTX</b>		All	recalls <b>x</b> and <b>y</b> .
LBL	label	PRG	
LINF	...	DECM	Selects the linear curve fit model.
LJ	...	2, 8, d, h	

Name	Keys to press	Works in modes	Remarks
LN	LN	DECM	
LN1+X	X.FCN ...	DECM	
LOG	LOG	DECM	
LOGF	X.FCN ...	DECM	Selects the logarithmic curve fit model.
LOGY	LGy	DECM	Calculates the logarithm for base <i>y</i> .
LOG2	LG2	DECM	Calculates the logarithm for base 2.
MASKL	X.FCN ...	2, 8, d, h	
MASKR	X.FCN ...	2, 8, d, h	
MEAN	$\bar{x}$	DECM	
MOD	MOD	\a	
N	X.FCN ...	DECM	Recalls the # of data points accumulated. Necessary for basic statistics.
NAND	X.FCN ...	2, 8, d, h	
NBITS	X.FCN #B	2, 8, d, h	Like #B on HP-16C.
NOR	X.FCN ...	2, 8, d, h	
NOT	NOT	2, 8, d, h	
N!	X.FCN ...	DECM	The same result is obtained by <i>x</i> !
OCTM	8	\a	Sets octal mode.
OFF	OFF	All	
	X.FCN ...	PRG	
ON		Calc. off	
	X.FCN ...	PRG	
OR	OR	2, 8, d, h	
PAUSE	PAUSE	PRG	
PERM	Py.x	DECM	









Name	Keys to press	Works in modes	Remarks
PGOFF	<b>h</b> <b>PRGM</b>	PRG	Toggles programming mode.
PGON		\PRG	
PI	<b>h</b> <b>π</b>	DECM	As long as no reassignment took place.
	<b>F3</b>	DECM	
PWRF	<b>h</b> <b>X.FCN</b> ...	DECM	Selects the power curve fit model.
QZ	<b>f</b> <b>Qz</b>	DECM	
RAD	<b>h</b> <b>RAD</b>	DECM	
RAN	<b>h</b> <b>RN#</b>	\a	
<i>RCL</i>	<b>RCL</b> reg	\h	<b>CPX</b> <b>RCL</b> recalls the register specified and the next adjacent register to <b>x</b> and <b>y</b> .
	<b>f</b> <b>RCL</b> reg	h	
RCLWS	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	Recalls the word size set.
<i>RCL+</i>	<b>RCL</b> <b>+</b> reg	\h (needs <b>f</b> in hex mode)	<b>CPX</b> <b>RCL</b> recalls the register specified and the next adjacent register to <b>x</b> and <b>y</b> .
<i>RCL-</i>	<b>RCL</b> <b>-</b> reg		
<i>RCLx</i>	<b>RCL</b> <b>x</b> reg		
<i>RCL/</i>	<b>RCL</b> <b>/</b> reg		
RDN	<b>R↓</b>	All	
RDX,	<b>h</b> <b>./,</b>	. selected	Toggles radix mark.
RDX.		, selected	
RL	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
RLC	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
RLCN	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
RLN	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
<i>RND</i>	<b>g</b> <b>RND</b>	DECM	
RR	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	

Name	Keys to press	Works in modes	Remarks
RRC	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
RRCN	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
RRN	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
RTN	<b>g</b> <b>RTN</b>	PRG	
RUP	<b>h</b> <b>R↑</b>	All	
R/S	<b>R/S</b>	All	
SB	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
SCI	<b>h</b> <b>SCI</b> #	DECM	
SDEV	<b>g</b> <b>S</b>	DECM	
SERR	<b>h</b> <b>X.FCN</b> ...	DECM	Calculates $\frac{SDEV}{\sqrt{N}}$ .
SEED	<b>STO</b> <b>h</b> <b>RN#</b>	DECM	
SF	<b>h</b> <b>SF</b>	All	
SHOW	<b>h</b> <b>SHOW</b>	All	
SIGMA	<b>h</b> <b>X.FCN</b> ...	DECM	Calculates $SDEV \cdot \sqrt{\frac{N}{N-1}}$ .
<b>SIN</b>	<b>f</b> <b>SIN</b>	DECM	
<b>SINH</b>	<b>f</b> <b>HYP</b> <b>SIN</b>	DECM	
SL	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
SOLVE	<b>f</b> <b>SOLVE</b> label	DECM	
<b>SQRT</b>	<b>f</b> <b>√x</b>	DECM	
SR	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
SST	<b>▼</b>	All	
STATUS	<b>h</b> <b>STATUS</b>	All	

Name	Keys to press	Works in modes	Remarks
STO	<b>STO</b> reg	\h	<b>CPX</b> <b>STO</b> stores <b>x</b> and <b>y</b> into the register specified and the next adjacent register.
	<b>f</b> <b>STO</b> reg	h	
STOP	<b>R/S</b>	PRG	
STO+	<b>STO</b> <b>+</b> reg	\h (needs <b>f</b> in hex mode)	<b>CPX</b> <b>STO</b> stores <b>x</b> and <b>y</b> into the register specified and the next adjacent register.
STO-	<b>STO</b> <b>-</b> reg		
STOx	<b>STO</b> <b>x</b> reg		
STO/	<b>STO</b> <b>/</b> reg		
STOWS	<b>h</b> <b>X.FCN</b> <b>WS</b> bits	b, 8, d, h	Sets the word size.
SUM	<b>RCL</b> <b>Σ+</b>	DECM	
TAN	<b>f</b> <b>TAN</b>	DECM	
TANH	<b>f</b> <b>HYP</b> <b>TAN</b>	DECM	
UNDO	<b>f</b> <b>UNDO</b>	All	
UNSIGN	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
VIEW	<b>h</b> <b>VIEW</b> reg	All	
WMEAN	<b>h</b> <b>X.FCN</b> <b>WMN</b>	DECM	Weighted mean.
XEQ	<b>XEQ</b> label	All	
XNOR	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
XOR	<b>f</b> <b>XOR</b>	2, 8, d, h	
X!	<b>h</b> <b>!</b>	DECM	
X.FCN	<b>h</b> <b>X.FCN</b>	DECM	Calls the catalog of extra real functions and sets alpha mode to allow for keying in names of operations. <b>▲</b> and <b>▼</b> browse the catalog. Choose the operation displayed by pressing <b>ENTER</b> <b>↑</b> . – <b>CPX</b> <b>h</b> <b>X.FCN</b> calls the catalog of extra complex functions.
		2, 8, d, h	Calls the catalog of extra integer functions. See above for more.

Name	Keys to press	Works in modes	Remarks
X<>	<b>h</b> <b>x↔</b> reg	All	<b>CPX</b> <b>h</b> <b>x↔</b> exchanges <b>x</b> and <b>y</b> with the register specified and the next adjacent register.
X<>Y	<b>x↔y</b>	All	<b>CPX</b> <b>x↔y</b> exchanges <b>x</b> and <b>y</b> with <b>z</b> and <b>t</b> .
X< ?	<b>f</b> <b>x&lt;?</b> arg	\a	See the table above showing the opportunities for "arg".
X≤ ?	<b>g</b> <b>x≤?</b> arg		
X= ?	<b>f</b> <b>x=?</b> arg		
X≠ ?	<b>g</b> <b>x≠?</b> arg		
X> ?	<b>f</b> <b>x&gt;?</b> arg		
X≥ ?	<b>g</b> <b>x≥?</b> arg		
X↑2	<b>g</b> <b>x<sup>2</sup></b>	DECM	
Y↑X	<b>g</b> <b>y<sup>x</sup></b>	DECM	
	<b>F2</b>	DECM	As long as no reassignment took place.
ZP	<b>g</b> <b>zP</b>	DECM	
0 ... 9	<b>0</b> ...	All	
1/X	<b>f</b> <b>1/x</b>	DECM	
	<b>F1</b>	DECM	As long as no reassignment took place.
1CPL	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
2CPL	<b>h</b> <b>X.FCN</b> ...	2, 8, d, h	
2↑X	<b>g</b> <b>2<sup>x</sup></b>	DECM	
10↑X	<b>g</b> <b>10<sup>x</sup></b>	DECM	
[.] or [,]	<b>.</b>	DECM	Inserts the radix mark as selected.
		a	Inserts a point (if RDX.) or comma (if RDX,).
	<b>h</b> <b>./,</b>	a	Inserts a comma (if RDX.) or point (if RDX,).
[ ] or [/]	<b>.</b>	/c	

Name	Keys to press	Works in modes	Remarks
[°], ['] or ["]		HMS	
+		\a	
-		\a	
x		\a	
/		\a	
+/-		\a	
//		DECM	Calculates $1/(1/x + 1/y)$ .
%		DECM	
%CH		DECM	
$\Sigma+$		DECM	
$\Sigma-$		DECM	
$\Sigma\text{LN}X$	...	DECM	Recalls the respective sum. These sums are necessary for the other curve fitting models beyond pure linear. See below for more.
$\Sigma\text{LN}XY$	...		
$\Sigma\text{LN}X^2$	...		
$\Sigma\text{LN}Y$	...		
$\Sigma\text{LN}Y^2$	...		
$\Sigma X$	...		
$\Sigma XY$	...	DECM	Recalls the respective sum. These sums are necessary for basic statistics and linear curve fitting. Calling them by name greatly enhances readability of programs.  Proposal: These 11 statistical sums (in total) shall be stored in registers 88 through 99.
$\Sigma X^2$	...		
$\Sigma Y$	...		
$\Sigma Y^2$	...		
$\rightarrow\text{DEG}$			
$\rightarrow\text{HMS}$		DECM	
$\rightarrow\text{HR}$		HMS	
$\rightarrow\text{POL}$		DECM	

Name	Keys to press	Works in modes	Remarks
→RAD	 	DECM	
→REC	 	DECM	
	 	2	Shift the display window like in HP-16C.
	 		

Edition	Date	Remarks
1	9.12.08	Start
1.1	15.12.08	Added the table of indicators; added NAND, NOR, XNOR, RCLWS, STOWS, //, N, SERR, SIGMA, < and >; deleted HR, INPUT, 2 flag commands, and 2 conversions; extended explanations for addressing and COMPLEX & ...; put XOR on the keyboard; corrected errors.
1.2		