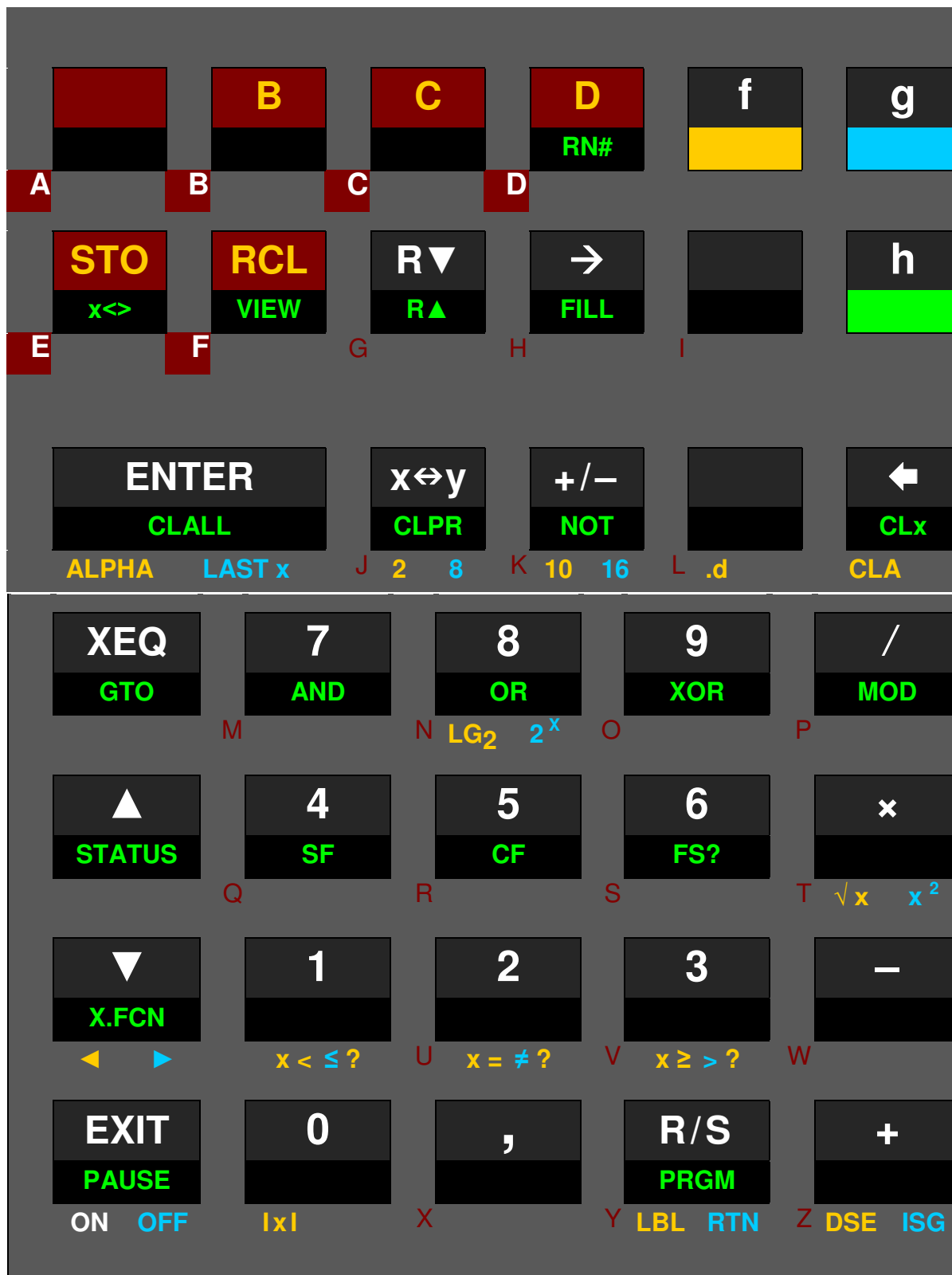Keyboard layout:

- CPX may be combined with +, -, x, /, ±, x^2, √x, 1/x, //, !, Γ, π, IxI, RND, as well as e.g. (HYP) SIN, COS, TAN, logs and their inverses. See the index for more.
- Modes are H.MS, 2, 8, 10, 16, .d, b/c, FIX, SCI, ENG, DEG, RAD, GRAD.
- → is combined with H, H.MS, DEG, RAD, GRAD, 2, 8, 10, 16 for conversions.
- The keys B, C, and D immediately call the respective user programs if existent.

	B	C	D	f	g
A	B	C	D RN#		

STO x<>	RCL VIEW	R▼ R▲	→ FILL		h
E	F	G	H	I	

ENTER CLALL	x↔y CLPR	+/− NOT		← CLx
ALPHA LAST x	J 2 8	K 10 16	L .d	CLA

XEQ GTO	7 AND	8 OR	9 XOR	/ MOD
M	N LG$_2$ 2^X	O	P

▲ STATUS	4 SF	5 CF	6 FS?	×
Q	R	S	T √x x^2

▼ X.FCN	1	2	3	−
◄ ►	x < ≤ ?	U x = ≠ ?	V x ≥ > ?	W

EXIT PAUSE	0	,	R/S PRGM	+
ON OFF	IxI	X	Y LBL RTN	Z DSE ISG

Active operations in <u>hexadecimal</u> mode. ➡ is for addressing only (see below). The primary functions of the top left 6 keys are numeric input, so their default primary functions are accessed using f-shift. – In the other integer modes, the active keyboard will look alike, but the primary functions of the top left 6 keys will stay as they are in DECM.
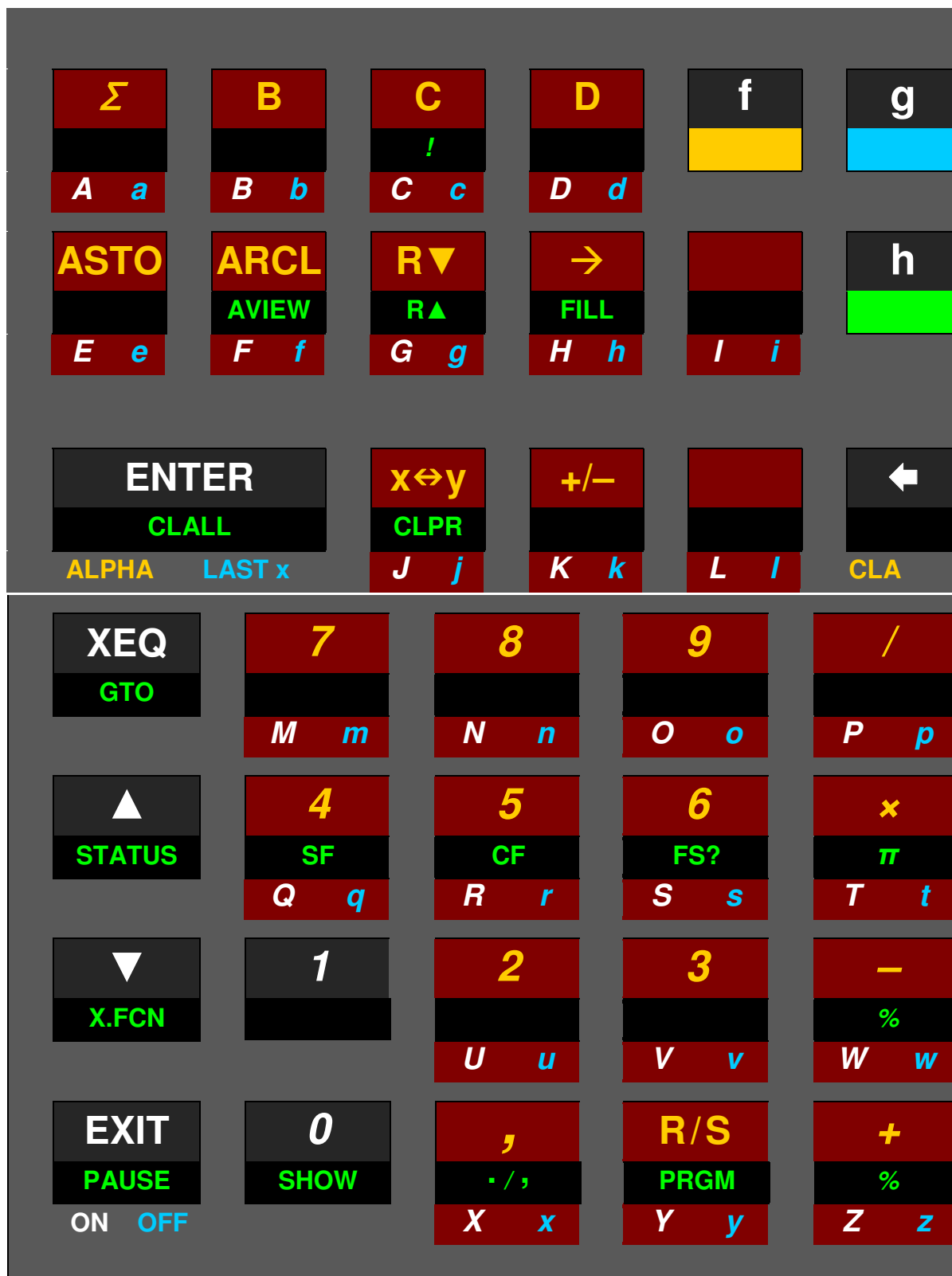
Active operations in <u>alpha</u> mode. ➡ is for addressing only (see below). The primary function of most keys will be inserting a letter. The basic arithmetic keys, Σ, π, %, and "!" will do so, too. To reach the default primary functions, f-shift will be necessary wherever a letter stands next to a key. PAUSE will insert a space.

ADDRESSING REGISTERS

1	User input	`x= ?` or any of the other comparisons				`RCL`, `STO`, `VIEW`, `x⇄`, `DSE`, `ISG`, `DSZ`, `ISZ`, `SF`, `CF`, `FS?` etc., `FIX`, `SCI`, `ENG`, `SB#` and the other commands of type `...#`		
	Display	**OP _** (e.g. `x > _`) Alpha mode is set.				**OP _** (e.g. `RCL _`)		
2	User input [1]	`0` or `1`	`Y`, `Z`, `T`, or `L`	`ENTER↑`	`ENTER↑` [2]	*Number of register or flag or bit or decimals* [3]		`→`
	Display	**OP 0** e.g. `x ≤ 0`	**OP x** e.g. `x ≥ y`	**OP r _**	**OP s _** Alpha mode is set.	**OP nn** e.g. `SF 15`		**OP → _**
						(indirect addressing)		
3	User input			*Register no.* [3]	`X`, `Y`, `Z`, `T`, or `L`	`ENTER↑`		*Register number* [3]
	Display	Compare **x** with the number in register **23**.		**OP r nn** e.g. `x ≠ r23`	**OP s x** e.g. `STO ≤Z`	**OP →s _** Alpha mode is set.		**OP → nn** e.g. `SCI →03`
4	User input				Store **x** on stack level **z**.	`X`, `Y`, `Z`, `T`, or `L`		Choose scientific number display with the number of decimals specified in register **03**.
	Display				Show the content of the register where **LASTx** is pointing to.	**OP →s x** e.g. `VIEW →sL`		

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

[2] For **RCL**, **STO**, **VIEW**, and **x<>** only.

[3] Register and flag numbers may be `0` `0` … `9` `9`, bit numbers `0` `1` … `6` `4`, number of decimals `0` … `9` . For numbers <10, you may key in e.g. `5` `ENTER↑` instead of `0` `5` .

ADDRESSING LABELS

| 1	User input	(GTO) or (XEQ) , (LBL), (SOLVE), (INTEG) [4]		
	Display	OP "_ (e.g. GTO _) Alpha mode is set		
2	User input	*Label* + (ENTER↑) [5]	(ENTER↑) (→)	
	Display	OP "*name*" e.g. SLV"STF"	OP → _	
		Solve the function **STF** (with STF keyed in).	(indirect addressing)	
3	User input		(ENTER↑)	*Register number*
	Display		OP →s _ Alpha mode is set.	OP → *nn* e.g. XEQ →44
4	User input		(X) , (Y) , (Z) , (T) , or (L)	Execute the routine which's label is in register **44**.
	Display	Integrate the function which's label is on stack level **y**.	OP →s *x* e.g. INT →sY	

---

[4] **SOLVE** and **INTEG** will be displayed as SLV and INT, respectively. No indirect addressing with **LBL**.

[5] 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.

Indicator	*INPUT* a	b	d	h	8	*STO* PRG
Set by operation	AON ENTER X.FCN	BINM	IDECM	HEXM	OCTM	PRGON
Cleared by operation	AOFF ENTER	DECM FRACM HEXM IDECM OCTM	BINM DECM FRACM HEXM OCTM	BINM DECM FRACM IDECM OCTM	BINM DECM FRACM HEXM IDECM	PRGOFF

Indicator	*360*	g	HMS	*RAD*	∕c
Set by operation	DEG	GRAD	HMSM >HMS TIMER	RAD ACOSH ASINH ATANH	FRACM, 2nd ⌊.⌋ in input (\HMS)
Cleared by operation	GRAD RAD ACOSH ASINH ATANH	DEG RAD ACOSH ASINH ATANH >HR	DECM >HR	DEG GRAD >HR	BINM DECM IDECM HEXM OCTM

INPUT, STO, 360, and RAD are annunciators. Outside integer modes, everything else is indicated in the text line. An active timer application is signaled by T, the different date modes by D.MY or M.DY. Defaults Y.MD and DECM are not indicated.

Within integer modes, word size and complement setting are shown in the text line. The exponent is used for the further indications: its first digit shows the base, a "c" signals a carry bit set, an "o" an overflow.

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, the operations will work like on the HP-42S, special bit and integer functions like on the HP-16C. Functions available here for the first time on an RPN calculator are highlighted. If no parameters are specified though required, they will be taken from the stack. Modes are abbreviated by their indicators. There, a backslash stands for "all but", so e.g. ABS works in all modes but alpha.

Name	Keys to press	Works in modes	Remarks
A … F	[A] etc.	h	The top left 6 keys will work for numeric input in hexadecimal mode by default.
ABS	[f] [|x|]	\a	[CPX] [f] [|x|] calculates $\sqrt{x^2 + y^2}$.
ACOS	[g] [COS⁻¹]	DECM, H.MS	
ACOSH	[g] [HYP⁻¹] [COS]	DECM	
AIP	[h] [X.FCN] **AIP**	\a	
ALL	[h] [X.FCN] **ALL**	DECM	
ALENG	[h] [X.FCN] **ALENG**	\a	
AND	[h] [AND]	b, 8, d, h	
AOFF	[f] [ALPHA] / [ENTER↑]	a	Toggles alpha mode.
AON	[f] [ALPHA] / [ENTER↑]	\a / *)	*) Please see the table for register addressing above for details.
ARCL	[f] [RCL] *reg*	a	See opportunities for *reg* in the table above.
AROT	[h] [X.FCN] **AROT**	\a	
ASHF	[h] [X.FCN] **ASHF**	All	
ASIN	[g] [SIN⁻¹]	DECM, H.MS	
ASINH	[g] [HYP⁻¹] [SIN]	DECM	
ASR	[h] [X.FCN] **ASR**	b, 8, d, h	Shifts *x* right 1 bit like in HP-16C.
ASR#	[h] [X.FCN] **ASR#** *n*	b, 8, d, h	Will work like *n* consecutive ASRs. See the opportunities for *n* in the table above.

Name	Keys to press	Works in modes	Remarks
ASRN	**h** **X.FCN** **ASRn**	b, 8, d, h	Shifts *y* right I*x*I bits.
ASTO	**f** **STO** *reg*	a	See opportunities for *reg* in the table above.
ATAN	**g** **TAN⁻¹**	DECM, H.MS	
ATANH	**g** **HYP⁻¹** **TAN**	DECM	
ATOX	**h** **X.FCN** **A→X**	\a	
AVIEW	**h** **SHOW**	a	Or **h** **VIEW**
	h **X.FCN** **AVIEW**	\a	
A0	**h** **L.R.**	DECM	Calculates the parameters of the fit curve (through the data points accumulated) according to the model selected. In the linear model, A0 is the intercept and A1 the slope of the regression line.
A1	**h** **L.R.** **x⩾y**		
BASE+	**+**	b, 8, d, h	
BASE−	**−**		
BASE×	**x**		
BASE/	**/**		
BASE+/−	**+/−**		
BC?	**h** **X.FCN** **BC?**	b, 8, d, h	Works in analogy to BS?.
BC?#	**h** **X.FCN** **BC?#** *n*	b, 8, d, h	Tests the *n*-th bit in *x*. See the opportunities for *n* in the table above.
BETA	**h** **X.FCN** **β(x,y)**	DECM	Calculates Euler's Beta function.
BINM	**f** **2**	\a	Sets binary integer mode.
BITS	**h** **X.FCN** **BITS**	b, 8, d, h	Counts bits set like **#B** on HP-16C.
BST	**▲**	PRG	Go 1 step back in program memory.
		\PRG	Go 1 step back without executing this step.
BS?	**h** **X.FCN** **BS?**	b, 8, d, h	Works like **B?** in HP-16C.
BS?#	**h** **X.FCN** **BS?#** *n*	b, 8, d, h	Works in analogy to BC?#.

Name	Keys to press	Works in modes	Remarks
CAPP	**h** `X.FCN` **CAPP** *char*	All	Switches to alpha mode for the input of 1 char, appends this to the alpha register, and returns to the mode set before. Eventually , this equals the sequence AON *char* AOFF.
CB	**h** `X.FCN` **CB**	b, 8, d, h	Clears the *x*-th bit in *y* like in HP-16C.
CB#	**h** `X.FCN` **CB#** *n*	b, 8, d, h	Clears the *n*-th bit in *x*. See the opportunities for *n* in the table above.
CEIL	**h** `X.FCN` **CEIL**	DECM	Computes largest integer ≤ *x*.
CF	**h** `CF` *n*	All	See opportunities for *n* in the table above.
CHS	`+/−`	\a	`CPX` `+/−` changes signs of *x* and *y*.
CLA	**f** `CLA`	All	Clears the alpha register.
CLALL	**h** `CLALL`	\PRG	Global clear after confirmation.
CLPR	**h** `CLPR`	PRG	Clears current program after confirmation.
CLRG	**h** `X.FCN` **CLRG**	All	Clears all general purpose registers.
CLSTK	`0` **h** `FILL`	All	
CLX	**h** `CLx`	All	`CPX` **h** `CLx` clears **x** and **y**.
CLΣ	**h** `CLΣ`	DECM	
COMB	**f** `Cx,y`	DECM	
COMPLEX	`CPX` ...	DECM	Indicates complex operations, acting on *x* and *y*, where **x** contains the real part and **y** the imaginary of the complex number. This key may be combined with any function which's name is printed in *italics* in this table.
CONJ	**h** `CONJ`	DECM	Changes the sign of *y*.
CORR	**g** `r`	DECM	
COS	**f** `COS`	DECM, H.MS	
COSH	**f** `HYP` `COS`	DECM	
DATE	**h** `X.FCN` **DATE**	DECM	Adds a number of days in *x* on a date in *y* and displays the resulting date including the respective weekday (Sunday = 7). This function works like in HP-12C.

Name	Keys to press	Works in modes	Remarks
DDAYS	**h** X.FCN ΔDAYS	DECM	Calculates the number of days between 2 dates **x** and **y**. Function works like in 12C.
DECM	**f** .d	\a	H.MS data in **x** will be converted to decimal.
DEG	**g** DEG	DECM	
DENMAX	**h** X.FCN /c *n*	DECM	Sets the maximum denominator for fractions.
DSE	**f** DSE *reg*	PRG	See opportunities for *reg* in the table above.
DSZ	**h** X.FCN DSZ	PRG	
D.MY	**h** X.FCN D.MY	DECM	Sets the format for date calculations.
D→R	**h** X.FCN D→R	DECM	Assumes **x** containing radians and converts them to degrees. Mode is kept constant.
EEX	**E**	DECM	
EMGAM	**h** X.FCN γ	DECM	Recalls the Euler-Mascheroni constant.
ENG	**f** ENG *n*	DECM	See opportunities for *n* in the table above.
ENTER↑	ENTER↑	All	
EXIT	EXIT	All	Exits X.FCN and any other menus or functions with pending input.
EXPF	**h** X.FCN EXPF	DECM	Selects the exponential curve fit model.
E↑X	**g** *eˣ*	DECM	
FB	**h** X.FCN FB	b, 8, d, h	Inverts ("flips") the **x**-th bit in **y**.
FB#	**h** X.FCN FB# *n*	b, 8, d, h	Flips the *n*-th bit in **x.** See the opportunities for *n* in the table above.
FCSTX	**h** X.FCN x^	DECM	FCSTX (*FCSTY*) predicts x (*y*) for a given y (*x*) according to the curve fit model chosen. See A0 and A1 for more.
FCSTY	**f** ŷ		
FC?	**h** X.FCN FC? *n*	All	See opportunities for *n* in the table above.
FC?C	**h** X.FCN FC?C *n*		
FF	**h** X.FCN FF *n*	All	Inverts ("flips") the flag specified. See the opportunities for *n* in the table above.
FILL	**h** FILL	All	Copies **x** in **y**, **z**, and **t**.

Name	Keys to press	Works in modes	Remarks
FIX	**f** (FIX) *n*	DECM	See opportunities for *n* in the table above.
FLOAT	**f** (.d)	\a	Works like DECM.
FLOOR	**h** (X.FCN) **FLOOR**	DECM	Computes the smallest integer ≥ *x*.
FP	**g** (FP)	DECM	
FRACM	**g** (b/c)	DECM	Sets fraction mode like in HP-32SII.
FS?	**h** (FS?) *n*	All	See opportunities for *n* in the table above.
FS?C	**h** (X.FCN) **FS?C** *n*		
GAMMA	**h** (X.FCN) Γ(x)	DECM	
GRAD	**g** (GRAD)	DECM	
GSB	**h** (XEQ) *label*	PRG	Works like XEQ.
GTO	**h** (GTO) *label*	PRG	Like in HP32S
		\PRG	
	h (GTO) (.) *label*	\PRG	
	h (GTO) (.) (.)		
HEXM	**g** (16)	\a	Sets hexadecimal integer mode.
HMSM	**g** (H.MS)	DECM	Sets H.MS mode.
HMS+	(+)	H.MS	
HMS−	(−)	H.MS	
IDECM	**f** (10)	\a	Sets integer decimal mode.
INTEG	**h** (INTEG) *label*	DECM	Parameters will be transferred like in 15C.
IP	**f** (IP)	DECM	
ISG	**g** (ISG) *reg*	PRG	See opportunities for *reg* in the table above.
ISZ	**h** (X.FCN) **ISZ**	PRG	
LASTX	**g** (LASTx)	All	(CPX) **g** (LASTx) recalls **x** and **y**.
LBL	**f** (LBL) *label*	PRG	

Name	Keys to press	Works in modes	Remarks
LINF	**h** X.FCN **LINF**	DECM	Selects the linear curve fit model.
LJ	**h** X.FCN **LJ**	b, 8, d, h	
LN	**f** LN	DECM	
LNBETA	**h** X.FCN **LN(β)**	DECM	Calculates the logarithm of BETA or GAMMA, respectively. See these functions.
LNGAMMA	**h** X.FCN **LN(Γ)**		
LOG	**f** LOG	DECM	
LOGF	**h** X.FCN **LOGF**	DECM	Selects the logarithmic curve fit model.
LOGY	**f** LGy	DECM	Calculates the logarithm for base *y*.
LOG2	**f** LG2	\a	Calculates the logarithm for base 2.
MASKL	**h** X.FCN **MASKL**	b, 8, d, h	
MASKL#	**h** X.FCN **MASKL#** *n*	b, 8, d, h	See opportunities for *n* in the table above.
MASKR	**h** X.FCN **MASKR**	b, 8, d, h	
MASKR#	**h** X.FCN **MASKR#** *n*	b, 8, d, h	See opportunities for *n* in the table above.
MAX	**h** X.FCN **MAX**	\a	Returns the maximum of *x* and *y*.
MEAN	**f** x̄	DECM	
MIN	**h** X.FCN **MIN**	\a	Returns the minimum of *x* and *y*.
MIRROR	**h** X.FCN **MIRROR**	b, 8, d, h	Reflects bit patterns (e.g. 000101 → 101000)
MOD	**h** MOD	\a	
M.DY	**h** X.FCN **M.DY**	DECM	Sets the format for date calculations.
N	**h** X.FCN **N**	DECM	Recalls the number of data points accumulated. Necessary for basic statistics.
NAND	**h** X.FCN **NAND**	b, 8, d, h	
NOP	**h** X.FCN **NOP**	PRG	
NOR	**h** X.FCN **NOR**	b, 8, d, h	

Name	Keys to press	Works in modes	Remarks
NOT	**h** `NOT`	b, 8, d, h	
OCTM	**g** `8`	\a	Sets octal integer mode.
OFF	**g** `OFF`	\PRG	
	h `X.FCN` **OFF**	PRG	
ON	`ON`	Calc. off	
	h `X.FCN` **ON**	PRG	
OR	**h** `OR`	b, 8, d, h	
PAUSE	**h** `PAUSE`	PRG	
PERM	**g** `Py,x`	DECM	
PGOFF	**h** `PRGM`	PRG	Toggles programming mode.
PGON		\PRG	
PI	**h** `π`	DECM	`CPX` **h** `π` puts π in **y** and clears **x** .
	`D`	DECM	As long as no reassignment took place.
PROMPT	**h** `X.FCN` **PROMPT**	PRG	
PWRF	**h** `X.FCN` **PWRF**	DECM	Selects the power curve fit model.
QZ	**f** `Qz`	DECM	Like in HP-32E and HP-21S
RAD	**g** `RAD`	DECM	
RAND	**h** `RN#`	\a	
RCL	`RCL` *reg*	\h, \a	See RCL+ for more details.
	f `RCL` *reg*	h	
RCLWS	**h** `X.FCN` **RCLWS**	b, 8, d, h	Recalls the word size set.
RCL+	`RCL` `+` *reg*	\h, \a (needs **f** in hex mode)	See opportunities for *reg* in the table above. `CPX` `RCL` recalls the register specified and the next adjacent register to **x** and **y**.
RCL−	`RCL` `−` *reg*		
RCL×	`RCL` `×` *reg*		
RCL/	`RCL` `/` *reg*		

Name	Keys to press	Works in modes	Remarks
RDN	⟨R↓⟩	All	
RDX, **RDX.**	**h** ⟨./,⟩	DECM	Toggles radix mark.
REAL	**f** ⟨.d⟩	\a	Works like DECM.
RJ	**h** ⟨X.FCN⟩ **RJ**	b, 8, d, h	Works in analogy to LJ.
RL	**h** ⟨X.FCN⟩ **RL**	b, 8, d, h	
RL#	**h** ⟨X.FCN⟩ **RL#** *n*	b, 8, d, h	Will work like *n* consecutive RLs. See the opportunities for *n* in the table above.
RLC	**h** ⟨X.FCN⟩ **RLC**	b, 8, d, h	
RLC#	**h** ⟨X.FCN⟩ **RLC#** *n*	b, 8, d, h	Will work like *n* consecutive RLCs. See the opportunities for *n* in the table above.
RLCN	**h** ⟨X.FCN⟩ **RLCn**	b, 8, d, h	
RLN	**h** ⟨X.FCN⟩ **RLn**	b, 8, d, h	
RND	**g** ⟨RND⟩	DECM	
RR	**h** ⟨X.FCN⟩ **RR**	b, 8, d, h	
RR#	**h** ⟨X.FCN⟩ **RR#** *n*	b, 8, d, h	Will work like *n* consecutive RRs. See the opportunities for *n* in the table above.
RRC	**h** ⟨X.FCN⟩ **RRC**	b, 8, d, h	
RRC#	**h** ⟨X.FCN⟩ **RRC#** *n*	b, 8, d, h	Will work like *n* consecutive RRCs. See the opportunities for *n* in the table above.
RRCN	**h** ⟨X.FCN⟩ **RRCn**	b, 8, d, h	
RRN	**h** ⟨X.FCN⟩ **RRn**	b, 8, d, h	
RTN	**g** ⟨RTN⟩	PRG	
RUP	**h** ⟨R↑⟩	All	
R/S	⟨R/S⟩	\PRG, \T	
		T	Starts/stops incrementing the timer.
R→D	**h** ⟨X.FCN⟩ **R→D**	DECM	Assumes **x** containing degrees and converts them to radians. Mode is kept constant.
SB	**h** ⟨X.FCN⟩ **SB**	b, 8, d, h	

Name	Keys to press	Works in modes	Remarks
SB#	**h** X.FCN **SB#** *n*	b, 8, d, h	See opportunities for *n* in the table above.
SCI	**f** SCI *n*	DECM	See opportunities for *n* in the table above.
SDEV	**g** s	DECM	
SERR	**h** X.FCN **SERR**	DECM	Calculates $\dfrac{SDEV}{\sqrt{N}}$.
SEED	STO **h** RN#	DECM	
SF	**h** SF *n*	All	See opportunities for *n* in the table above.
SGMNT	**h** X.FCN **SGMNT**	All	Sets sign-and-mantissa mode for integers.
SHOW	**h** SHOW	DECM	Shows the full mantissa.
		a	Shows the alpha register.
SIGMA	**h** X.FCN **σ**	DECM	Calculates $SDEV \cdot \sqrt{\dfrac{N}{N-1}}$.
SIN	**f** SIN	DECM, H.MS	
SINH	**f** HYP SIN	DECM	
SL	**h** X.FCN **SL**	b, 8, d, h	
SL#	**h** X.FCN **SL#** *n*	b, 8, d, h	Will work like *n* consecutive SLs. See the opportunities for *n* in the table above.
SLN	**h** X.FCN **SLn**	b, 8, d, h	Shifts *y* left l*x*l bits.
SOLVE	**h** SOLVE *label*	DECM	
SQRT	**f** √x̄	\a	
SR	**h** X.FCN **SR**	b, 8, d, h	
SR#	**h** X.FCN **SR#** *n*	b, 8, d, h	Will work like *n* consecutive SRs. See the opportunities for *n* in the table above.
SRN	**h** X.FCN **SRn**	b, 8, d, h	Shifts *y* right l*x*l bits.
SST	▼	PRG	Go 1 step forward in program memory.
		\PRG	Go 1 step forward executing this step.
STATUS	**h** STATUS	All	

Name	Keys to press	Works in modes	Remarks
STO	(STO) *reg*	\h, \a	See STO+ for more details.
	(f) (STO) *reg*	h	
STOP	(R/S)	PRG	
STO+	(STO) (+) *reg*	\h, \a (needs (f) in hex mode)	See opportunities for *reg* in the table above. (CPX) (STO) stores *x* and *y* into the register specified and the next adjacent register.
STO–	(STO) (−) *reg*		
STO×	(STO) (×) *reg*		
STO/	(STO) (/) *reg*		
SUM	(RCL) (Σ+)	DECM	
TAN	(f) (TAN)	DECM, H.MS	
TANH	(f) (HYP) (TAN)	DECM	
TIMER	(h) (TIMER) *reg*	\T	Enters the timer application (sets indicator T, and HMS if not set yet). See opportunities for *reg* in the table above. Clears the register specified.
TIMEX	(h) (TIMER)	T	Leaves the timer application (clears indicator T, but H.MS stays on).
UNDO	(f) (UNDO)	All	
UNSIGN	(h) (X.FCN) **UNSIGN**	All	
VIEW	(h) (VIEW) *reg*	All	See opportunities for *reg* in the table above.
W	(h) (X.FCN) **W(x)**	DECM	Calculates Lambert's W for a given x ≥ -1/e
WINV	(h) (X.FCN) **W** $^{-1}$	DECM	Inverts W, i.e. calculates x for a given W (≥ -1).
WMEAN	(h) (X.FCN) **WMEAN**	DECM	Calculates the weighted mean.
WSIZE	(h) (X.FCN) **WSIZE**	All	Sets the word size (up to 64 bits).
XEQ	(XEQ) *label*	PRG	Calls the respective subroutine.
		\PRG	Executes the respective program.
XNOR	(h) (X.FCN) **XNOR**	b, 8, d, h	
XOR	(h) (XOR)	b, 8, d, h	

Name	Keys to press	Works in modes	Remarks
XTOA	**h** X.FCN **X→A**	\a	
X!	**h** **!**	DECM	
X.FCN	**h** X.FCN	DECM	Calls the catalog of extra real functions and sets alpha mode to allow for keying in names of operations. ▲ and ▼ browse the catalog. Choose the operation displayed by pressing ENTER↑ . CPX **h** X.FCN calls the catalog of extra complex functions.
		b, 8, d, h	Calls the catalog of extra integer functions. See above for more.
X<>	**h** x≷ *reg*	All	See opportunities for *reg* in the table above. CPX **h** x≷ will exchange *x* and *y* with the register specified and the next adjacent register.
X<>Y	x≷y	All	Performs **Re <> Im** if a complex operation was executed before. CPX x≷y will exchange *x* and *y* with *z* and *t*.
X< ?	**f** x < ? *arg*	\a	
X≤ ?	**g** x ≤ ? *arg*		
X= ?	**f** x = ? *arg*		See opportunities for *arg* in the table above.
X≠ ?	**g** x ≠ ? *arg*		
X≥ ?	**f** x ≥ ? *arg*		
X> ?	**g** x > ? *arg*		
X↑2	**g** x²	\a	
Y.MD	**h** X.FCN **Y.MD**	DECM	Sets the format for date calculations.
Y↑X	**g** yˣ	DECM	
	C	DECM	As long as no reassignment took place.
ZETA	**h** X.FCN ζ(x)	DECM	Calculates Riemann's Zeta.
ZP	**g** zP	DECM	Like in **Q⁻¹** in HP-32E and z_P in HP-21S
0, 1	**0** , **1**	All	

Name	Keys to press	Works in modes	Remarks
2 … 7	⎡2⎤ …⎡7⎤	\b	
8, 9	⎡8⎤ , ⎡9⎤	\b, \8	
1/X	⎡f⎤ ⎡1/x⎤	DECM	
	⎡B⎤	DECM	As long as no reassignment took place.
1CPL	⎡h⎤ ⎡X.FCN⎤ **1CPL**	All	Like 1's complement in HP-16C.
2CPL	⎡h⎤ ⎡X.FCN⎤ **2CPL**	All	Like 2's complement in HP-16C.
2↑X	⎡g⎤ ⎡2ˣ⎤	\a	
10↑X	⎡g⎤ ⎡10ˣ⎤	DECM	
[.] or [,]	⎡.⎤	DECM	Inserts the radix mark as selected.
		a	Inserts a point if RDX. , else a comma.
	⎡h⎤ ⎡./,⎤	a	Inserts a comma if RDX. , else a point.
[ ] or [/]	⎡.⎤	/c	
[°]	⎡.⎤	H.MS	
+	⎡+⎤		
−	⎡−⎤		
×	⎡×⎤	DECM	
/	⎡/⎤		
+/−	⎡+/−⎤		
//	⎡g⎤ ⎡//⎤	DECM	Calculates $\left(\dfrac{1}{x}+\dfrac{1}{y}\right)^{-1}$.
%	⎡g⎤ ⎡%⎤	DECM	
%CH	⎡f⎤ ⎡Δ%⎤	DECM	
%+	⎡h⎤ ⎡%+⎤	DECM	Adds a markup of *x* % to *y*.
%−	⎡h⎤ ⎡%−⎤	DECM	Subtracts a discount of *x* % from *y*.
Σ+	⎡Σ+⎤	DECM	

Name	Keys to press	Works in modes	Remarks
Σ−	**h** Σ−	DECM	
ΣLNX	**h** X.FCN **ΣLNX**	DECM	Recalls the respective statistical sum. These sums are necessary for the other curve fitting models beyond pure linear. See below for more.
ΣLNXY	**h** X.FCN **ΣLNXY**		
ΣLNX2	**h** X.FCN **ΣLNX²**		
ΣLNY	**h** X.FCN **ΣLNY**		
ΣLNY2	**h** X.FCN **ΣLNY²**		
ΣX	**h** X.FCN **ΣX**	DECM	Recalls the respective statistical sum. These sums are necessary for basic statistics and linear curve fitting. Calling them by name greatly enhances readability of programs. These 11 statistical sums (in total) shall be stored in registers 88 through 99.
ΣXY	**h** X.FCN **ΣXY**		
ΣX2	**h** X.FCN **ΣX²**		
ΣY	**h** X.FCN **ΣY**		
ΣY2	**h** X.FCN **ΣY²**		
→BIN	➡ **f** 2	\a	Shows **x** in integer binary or decimal representation, respectively, until the next command is executed. Mode is kept constant.
→DEC	➡ **f** 10		
→DEG	➡ **g** DEG	DECM	Assumes **x** containing angles in current angular mode and converts them to degrees or gon, respectively. Mode is kept constant.
→GRAD	➡ **g** GRAD		
→HEX	➡ **g** 16	\a	Works like →BIN, but hexadecimal.
→HMS	➡ **g** H.MS	DECM	Assumes **x** containing *decimal* hours or degrees and converts them in the format HHH.MMSS.
→HR	➡ **f** H	H.MS	Takes the hours or degrees in **x** and converts them into decimal numbers.
→OCT	➡ **g** 8	\a	Works like →BIN, but octal.
→POL	**g** ▶P	DECM	Assumes **x** and **y** containing the coordinates x and y and converts them to r and ϑ.
→RAD	➡ **g** RAD	DECM, H.MS	Works like →DEG, but converts to radians.

Name	Keys to press	Works in modes	Remarks
→REC	**f** ▶R	DECM	Assumes **x** and **y** containing the coordinates r and ϑ and converts them to x and y.
	f ◀	b	Shift the display window like in HP-16C.
	g ▶		

Functions on the waitlist (personal priorities given by W):

1. **SOLVE** (included in the index already but not implemented yet)

2. **TIMER** (accuracy down to 0.1s is sufficient, we can't press keys more precisely – included in the index already but not implemented yet)

3. **INTEG**RATE (included in the index already but not implemented yet)

4. **T**, **CHISQ**, and **F** distributions (in this order) and their inverses (not in the index yet)

5. **Julian Calendar** calculations (not in the index yet)

6. Euler's **BETA** function, Lambert's **W** and Riemann's **ZETA** – **LNBETA** and **LNGAMMA** have even lower priority (included in the index already but may be dropped if bytes are needed)

7. Stranger additional flag and bit handling like **FC?C**, **BS?S**, etc. (doubt the use – the user may program this if wanted)

8. **x^y** (the user may reach this by pressing x<>y and y^x)

9. **%T** (i.e. y ENTER x / 100 *)

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	4.1.09	Added ASRN, CBC?, CBS?, CCB, SCB, FLOAT, MIRROR, SLN, SRN, >BIN, >DEC, >HEX, >OCT, BETA, D>R, DATE, DDAYS, D.MY, M.DY, Y.MD, CEIL, FLOOR, DSZ, ISZ, D>R, R>D, EMGAM, GSB, LNBETA, LNGAMMA, MAX, MIN, NOP, REAL, RJ, W and WINV, ZETA, %+ and %-; renamed the top left keys B, C, and D, and bottom left EXIT.
1.3	17.1.09	Added AIP, ALENG, ARCL, AROT, ASHF, ASTO, ATOX, XTOA, AVIEW, CLA, PROMPT (all taken from 42S), CAPP, FC?C, FS?C, SGMNT, and the …# commands; renamed NBITS to BITS and STOWS to WSIZE; specified the bit commands closer; deleted the 4 carry bit operations.