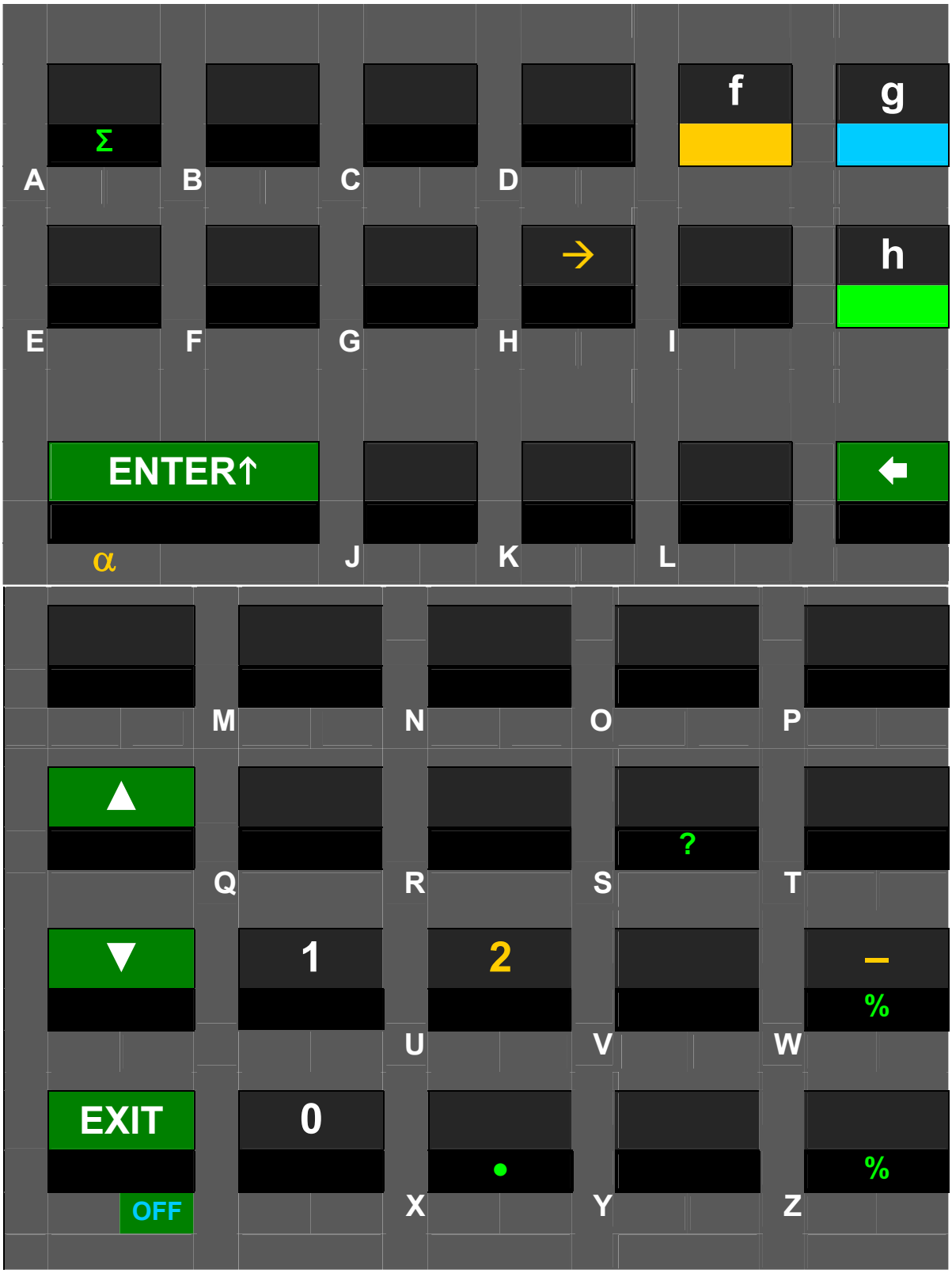

Keyboard layout (TIMER may be dropped for some reasons):

- CPX may be combined with +, -, x, /, ±, x^2, √x, 1/x, //, !, Γ, π, |x|, 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.


Active operations in hexadecimal 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 (except Σ+) will stay as they are in FLOAT.


Active operations in alpha mode. All labels not printed on green or red background here will insert the respective character. The alpha register is 15 bytes long. The primary function of most keys will be inserting the letter printed dark red bottom left of such a key on the original keyboard. For these keys, f-shift will be necessary to reach the function on key top. PAUSE will insert a space. ⬆ toggles upper and lower case.




Active operations in temporary alpha mode. These are necessary in the catalogues and during addressing while in this transient mode. The functions printed on green background allow for catalogue browsing, error recovery, and leaving. See regular alpha mode and the addressing tables for more.

## MEMORY

Stack

| | |
|---|---|
| **L** | **I** |

| |
|---|
| **T** |
| **Z** |
| **Y** |
| **X** |

**Display**

In complex arithmetic, stack registers **X** and **Y** contain the real and imaginary part of the first complex number, while **Z** and **T** carry the second. Stack register **I** takes the imaginary part of the last argument if a complex function is used. See LASTx.

As long as no complex function is used, **I** may be taken as another general purpose register.

The top 13 general purpose registers (87 through 99) take the statistical sums indicated as soon as [Σ+] is used.

General purpose registers

| |
|---|
| **00** |
| **01** |
| **02** |
| **…** |
| **86** |
| **87 Σ ln x** |
| **88 Σ ln² x** |
| **89 Σ ln y** |
| **90 Σ ln² y** |
| **91 Σ (ln x · ln y)** |
| **92 Σ (x ln y)** |
| **93 Σ (y ln x)** |
| **94 n** |
| **95 Σ x** |
| **96 Σ x²** |
| **97 Σ y** |
| **98 Σ y²** |
| **99 Σ xy** |

Program memory

| |
|---|
| **000** |
| **001** |
| **002** |
| **…** |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| **…** |
| **497** |
| **498** |
| **499** |

## ADDRESSING REGISTERS

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | User input | x= ? or any of the other comparisons | | | RCL, STO, aRCL, aSTO, VIEW, x≷, DSE, ISG, DSZ, ISZ, CF and the other flag commands, FIX, SCI, ENG, CB and many more bit commands | | | |
| | Display | OP _ (e.g. x > _ )<br>Alpha mode is on. | | | OP _ (e.g. RCL _ )<br>Alpha mode is off. | | | |
| 2 | User input [1] | 0 or 1 | Y, Z, T, L, or I | ENTER↑ closes alpha. | X, Y, Z, T, L, or I [2] | *Number of register or flag or bit or decimals* [3] | → | |
| | Display | OP *n*<br>e.g. x ≤ 0 | OP *x*<br>e.g. x ≥ y | OP r _ | OP s *x*<br>e.g. STO sZ | OP *nn*<br>e.g. SF 15 | OP → _<br>(indirect addressing) | |
| 3 | User input | | | *Register no.* [3] | Store ***x*** on stack level **z**. | X, Y, Z, T, L, or I | *Register number* [3] | |
| | Display | Compare ***x*** with the number in register **23**. | | OP r *nn*<br>e.g. x ≠ r23 | Show the content of the register where **LASTx** is pointing to. | OP →s *x*<br>e.g. VIEW →sL | OP → *nn*<br>e.g. SCI →03<br>Choose scientific number display with the number of decimals specified in register **03**. | |

[1] For **RCL** and **STO**, an arithmetic operator ( +, –, ×, / ) as well as MAX and MIN may precede step 2.

[2] For **RCL**, **STO**, **VIEW**, and **x<>** only. You may address the stack registers directly without switching to alpha mode before.

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

## ADDRESSING LABELS

| | | | | |
|---|---|---|---|---|
| 1 | User input | GTO or XEQ , LBL, SOLVE, INTEG [4] | | |
| | Display | OP "_ (e.g. GTO _ ) Alpha mode is on. | | |
| 2 | User input | *Label* + ENTER↑ [5] | ENTER↑ ➡ | |
| | Display | OP "*name*" e.g. SLV"STF" | OP → _ (indirect addressing, alpha mode is off) | |
| 3 | User input | Solve the function **STF** (with STF keyed in). | Any stack level, i.e. X , Y , Z , T , L, or I [6] | *Register number* |
| | Display | | OP →s *x* e.g. INT →sY | OP → *nn* e.g. XEQ →44 |
| | | | Integrate the function which's label is on stack level **y**. | Execute the routine which's label is in register **44**. |

[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.

[6] You may address the stack registers directly without switching to alpha mode before.

## 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** | αON [7] | BIN | DEC | HEX | OCT | PRGON |
| **Cleared by operation** | αOFF | DEC<br>FLOAT<br>FRACT<br>HEX<br>OCT | BIN<br>FLOAT<br>FRACT<br>HEX<br>OCT | BIN<br>DEC<br>FLOAT<br>FRACT<br>OCT | BIN<br>DEC<br>FLOAT<br>FRACT<br>HEX | PRGOFF |

| Indicator | *360* | g | HMS | *RAD* | /c |
|---|---|---|---|---|---|
| **Set by operation** | DEG | GRAD | H.MS<br>>H.MS<br>TIMER | RAD<br>ACOSH<br>ASINH<br>ATANH | FRACT,<br>2nd . in input ( \HMS) |
| **Cleared by operation** | GRAD<br>RAD<br>ACOSH<br>ASINH<br>ATANH | DEG<br>RAD<br>ACOSH<br>ASINH<br>ATANH | FLOAT<br>>H | DEG<br>GRAD | BIN<br>DEC<br>FLOAT<br>HEX<br>OCT |

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 FLOAT 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.

Any inputs are interpreted according to the modes set at input time.

[7] Alpha mode may be temporarily entered and left during addressing – see the tables above for more information. Alpha mode will also be entered opening the catalogues X.FCN, CONST, or CONV to allow for easy finding and selecting the item you are interested in. Keying in a character will jump to the first item starting with it, entering more will jump to the first item starting with this sequence, or anything following in the alphabet if this is not found. After executing the command chosen, the mode will return to the state it had before, except the command chosen did change the mode.

## 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. The characters necessary to specify the function in the catalogue are printed bold.

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 in the rightmost column, as are functions deviating from known ones carrying the same name. 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 in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| ***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 | |
| ALL | h X.FCN **ALL** | DECM | |
| **AND** | h AND | b, 8, d, h | |
| ***ASIN*** | g SIN$^{-1}$ | DECM, H.MS | |
| ***ASINH*** | g HYP$^{-1}$ SIN | DECM | |
| ASR | h X.FCN **ASR** *n* | b, 8, d, h | Will work like *n* consecutive ASRs on HP-16C. See the opportunities for *n* in the table above. |
| ***ATAN*** | g TAN$^{-1}$ | DECM, H.MS | |
| ***ATANH*** | g HYP$^{-1}$ TAN | DECM | |
| **BASE+** | + | b, 8, d, h | |
| **BASE–** | – | | |
| **BASE×** | × | | |
| **BASE/** | / | | |
| **BASE+/–** | +/– | | |
| 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. |
| BESTF | h X.FCN **BES**TF | DECM | Selects the best curve fit model according to the correlation found like in HP-42S. |
| *BETA* | h X.FCN **BET**A | DECM | Calculates Euler's Beta function. |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| **BIN** | f 2 | \a | Sets binary integer mode. |
| BS? | h X.FCN **BS**? *n* | b, 8, d, h | Works in analogy to "BC?". |
| B# | h X.FCN **B#** | b, 8, d, h | Counts bits set like **#B** on HP-16C. |
| **C …** | 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. The three dots will be replaced in the listing by the name of the function attached. |
| CB | h X.FCN **CB** *n* | b, 8, d, h | Clears the *n*-th bit in ***x***. See opportunities for *n* in the table above. |
| CEIL | h X.FCN **CE**IL | DECM | Computes largest integer ≤ ***x***. |
| **CF** | h CF *n* | \a | See opportunities for *n* in the table above. |
| CLALL | h X.FCN **CLA**LL | \PRG | Global clear. **Not programmable.** |
| **CLPR** | h CLPR | PRG | Clears current program after confirmation. **Not programmable.** |
| CLREG | h X.FCN **CLR**EG | \a | Clears all general purpose registers. |
| **CLSTK** | h CLST | \a | |
| ***CLx*** | h CLx | \a | CPX h CLx clears **X** and **Y**. |
| **CLα** | f CL α | All | Clears the alpha register like CLA in HP-42S. |
| **CLΣ** | g CLΣ | DECM | |
| **COMB** | f Cx,y | DECM | |
| **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 **DAT**E | DECM | Adds a number of days in ***x*** on a date in ***y*** and displays the resulting date including the day of week (Sunday = 7). This function works like in HP-12C. |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| DAY | h X.FCN **DAY** | DECM | Takes ***x*** as a date and returns an integer indicating the day of week (Sunday = 7) and displays the day of week in the dot matrix. |
| DBLR | h X.FCN **DBLR** | b, 8, d, h | Double precision commands like in HP-16C, but now for up to 128 bits. |
| DBL* | h X.FCN **DBL*** | | |
| DBL/ | h X.FCN **DBL/** | | |
| **DEC** | f 10 | \a | Sets integer decimal mode. |
| **DEG** | g DEG | DECM | |
| DENMAX | h X.FCN **DE**NMAX *n* | DECM | Sets the maximum denominator for fractions like **/c** does in HP-32SII. |
| DISP | h X.FCN **DI**SP *n* | DECM | Changes the number of decimals while keeping the mode (FIX, SCI, ENG). See opportunities for *n* in the table above. |
| **DSE** | f DSE *reg* | PRG | See opportunities for *reg* in the table above. |
| DSZ | h X.FCN **DSZ** *reg* | | |
| D.MY | h X.FCN **D.**MY | DECM | Sets the format for date calculations. |
| D→J | h X.FCN **D→J** | DECM | Assumes ***x*** is a date and converts it to a Julian day number. |
| D→R | h X.FCN **D→R** | DECM | Assumes **X** containing radians and converts them to degrees. Mode is kept constant. |
| **ENG** | f ENG *n* | DECM | See opportunities for *n* in the table above. |
| **ENTER↑** | ENTER↑ | All | |
| EXPF | h X.FCN **EX**PF | DECM | Selects the exponential curve fit model. |
| **$e^x$** | g $e^x$ | DECM | |
| *$e^x$ -1* | h X.FCN **$E^X$-1** | DECM | |
| 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 **FCS**TX | DECM | FCSTx (*FCSTy*) predicts a forecast x (*y*) for a given y (*x*) according to the curve fit model chosen. See L.R. for more. |
| **FCSTy** | f $\hat{y}$ | | |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| FC? | h X.FCN **FC?** *n* | \a | See opportunities for *n* in the table above. |
| FC?C | h X.FCN **FC?C** *n* | | |
| FF | h X.FCN **FF** *n* | \a | Inverts ("flips") the flag specified. See the opportunities for *n* in the table above. |
| **FILL** | h FILL | \a | Copies ***x*** in **Y**, **Z**, and **T**. |
| **FIX** | f FIX *n* | DECM | See opportunities for *n* in the table above. |
| **FLOAT** | f .d | \a | Works like DECM in HP-42S. Additionally, H.MS data in **X** will be converted to decimal. |
| FLOOR | h X.FCN **FL**OOR | DECM | Computes the smallest integer ≥ ***x***. |
| ***FP*** | g FP | DECM | |
| **FRACT** | g b/c | DECM | Sets fraction mode like in HP-32SII. |
| **FS?** | h FS? *n* | \a | See opportunities for *n* in the table above. |
| FS?C | h X.FCN **FS**?C *n* | | |
| *GAMMA* | h X.FCN **G**AMMA | DECM | |
| **GRAD** | g GRAD | DECM | |
| **GTO** | h GTO *label* | PRG | Like in HP32S. |
| | | \PRG | |
| | h GTO . *label* | \PRG | |
| | h GTO . . | | |
| **HEX** | g 16 | \a | Sets hexadecimal integer mode. |
| **H.MS** | g H.MS | DECM | Sets H.MS mode. |
| **H.MS+** | + | H.MS | |
| **H.MS–** | − | H.MS | |
| **INT** | h INTEG *label* | DECM | Integration 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 *reg* | | |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| J→D | h X.FCN **J**→D | DECM | Assumes ***x*** is a Julian day number and converts it to a date. |
| ***LASTx*** | g LASTx | \a | CPX g LASTx recalls ***I*** and ***i*** into **X** and **Y**. |
| **LBL** | f LBL *label* | PRG | |
| LEAP? | h X.FCN **LE**AP? | PRG | Checks if the integer part of ***x*** corresponds to a leap year. |
| LINF | h X.FCN **LI**NF | DECM | Selects the linear curve fit model. |
| LJ | h X.FCN **LJ** | b, 8, d, h | |
| ***LN*** | f LN | DECM | |
| *LNBETA* | h X.FCN **LNB**ETA | DECM | Calculates the natural logarithm of *β* or GAMMA, respectively. See these functions. |
| *LNGAMM* | h X.FCN **LNG**AMM | | |
| *LN1+* | h X.FCN **LN1**+ | DECM | |
| LOGF | h X.FCN **LO**GF | DECM | Selects the logarithmic curve fit model. |
| ***LOGy*** | f LOGy | DECM | Calculates the logarithm for base ***y***. |
| $LOG_{10}$ | f LG | DECM | |
| $LOG_2$ | f LOG2 | \a | Calculates the logarithm for base 2. |
| **LR** | h L.R. | DECM | Calculates the parameters of the fit curve (through the data points accumulated) according to the model selected. Returns A0 in **X** and A1 in **Y**. In the linear model, A0 is the intercept and A1 the slope of the regression line. |
| MASKL | h X.FCN MASKL *n* | b, 8, d, h | Work like MASKL and MASKR on HP-16C, but with the parameter following the command instead of taken from **X**. See the opportunities for *n* in the table above. |
| MASKR | h X.FCN MASKR *n* | | |
| MAX | h X.FCN **MAX** | \a | Returns the maximum of ***x*** and ***y***. |
| **MEAN** | f $\bar{x}$ | DECM | |
| MIN | h X.FCN **MIN** | \a | Returns the minimum of ***x*** and ***y***. |
| MIRROR | h X.FCN **MIR**ROR | b, 8, d, h | Reflects bit patterns (e.g. 000101 → 101000) |
| **MOD** | h MOD | \a | |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| M.DY | h X.FCN **M.**DY | DECM | Sets the format for date calculations. |
| n | h X.FCN **N** | DECM | Recalls the number of accumulated data points. Necessary for basic statistics. |
| NAND | h X.FCN **NA**ND | b, 8, d, h | |
| NOP | h X.FCN **NOP** | PRG | |
| NOR | h X.FCN **NOR** | b, 8, d, h | |
| **NOT** | h NOT | b, 8, d, h | |
| **OCT** | g 8 | \a | Sets octal integer mode. |
| OFF | h X.FCN **OF**F | PRG | |
| ON | h X.FCN **ON** | PRG | |
| **OR** | h OR | b, 8, d, h | |
| **PAUSE** | h PAUSE | PRG | Like PSE in HP42S. |
| **PERM** | g Py,x | DECM | |
| PROMPT | h X.FCN **PR**OMPT | PRG | |
| PWRF | h X.FCN **PW**RF | DECM | Selects the power curve fit model. |
| **Q(z)** | f Qz | DECM | Like in HP-32E and HP-21S |
| **RAD** | g RAD | DECM | |
| **RAND#** | h RN# | \a | Like RAN in HP42S. |
| ***RCL*** | RCL *reg* | \h, \a | See RCL+ for more details. |
| | f RCL *reg* | h | |
| RCLWS | h X.FCN **RC**LWS | b, 8, d, h | Recalls the word size set. |
| ***RCL+*** | RCL + *reg* | \h, \a (needs f in hex mode) | Calls the content of address *reg* , executes **OP** ***x*** on it and stores the result in **X**. RCL▲ (▼) takes the maximum (minimum) of the value in *reg* and ***x***. 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* | | |
| **RCL▲** | RCL ▲ *reg* | | |
| **RCL▼** | RCL ▼ *reg* | | |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| **RDX,**<br>**RDX.** | h ./, | DECM | Toggles radix mark. |
| RJ | h X.FCN **RJ** | b, 8, d, h | Works in analogy to LJ. |
| RL | h X.FCN **RL** *n* | b, 8, d, h | Will work like *n* consecutive RLs (RLCs) on HP-16C. See opportunities for *n* in the table above. |
| RLC | h X.FCN **RLC** *n* | | |
| RNDINT | h X.FCN **RN**DINT | DECM | Round ***x*** to next integer. ½ rounds to 1. |
| ***ROUND*** | g RND | DECM | Like RND in HP42S. |
| RR | h X.FCN **RR** *n* | b, 8, d, h | Will work like *n* consecutive RRs (RRCs) on HP-16C. See opportunities for *n* in the table above. |
| RRC | h X.FCN **RRC** *n* | | |
| **RTN** | g RTN | PRG | |
| **R/S** | R/S | \PRG, \T | |
| | | T | Starts/stops incrementing the timer. |
| **R↑** | h R↑ | \a | |
| **R↓** | R↓ | \a | |
| 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** *n* | b, 8, d, h | Sets the *n*-th bit in ***x***. 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 | |
| **SEED** | STO h RN# | DECM | |
| SERR | h X.FCN **SE**RR | DECM | Calculates $\frac{SDEV}{\sqrt{N}}$ . |
| **SF** | h SF *n* | \a | See opportunities for *n* in the table above. |
| SIGMA | h X.FCN **SIGM**A | DECM | Calculates $SDEV \cdot \sqrt{\frac{N}{N-1}}$ . |
| SIGN | h X.FCN **SIGN** | \a | |
| SIGNMT | h X.FCN **SIGNM**T | \a | Sets sign-and-mantissa mode for integers. |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| ***SIN*** | f SIN | DECM, H.MS | |
| ***SINH*** | f HYP SIN | DECM | |
| SL | h X.FCN **SL** *n* | b, 8, d, h | Will work like *n* consecutive SLs on HP-16C. See the opportunities for *n* in the table above. |
| **SLV** | h SOLVE *label* | DECM | |
| SR | h X.FCN **SR** *n* | b, 8, d, h | Will work like *n* consecutive SRs on HP-16C. See the opportunities for *n* in the table above. |
| ***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) | Calls the content of address *reg* , executes **OP *x*** on it and stores the result into said address. STO▲ (▼) takes the maximum (minimum) of the value in *reg* and ***x***. See opportunities for *reg* in the table above.<br>CPX STO stores ***x*** and ***y*** into the register specified and the next adjacent register. |
| ***STO–*** | STO − *reg* | | |
| ***STO×*** | STO × *reg* | | |
| ***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, \a | 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). |
| UNSIGN | h X.FCN **UNSIGN** | \a | |
| **VIEW** | h VIEW *reg* | \a | See opportunities for *reg* in the table above. |
| *W* | h X.FCN **W** | DECM | Calculates Lambert's W for a given x ≥ -1/e |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| *WINV* | h X.FCN **WI**NV | DECM | Inverts W, i.e. calculates x for a given W ( ≥ -1). |
| WMEAN | h X.FCN **WM**EAN | DECM | Calculates the weighted mean. |
| WSIZ | h X.FCN **WS**IZ *n* | \a | Works like WSIZE on HP-16C, but with the parameter following the command instead of taken from **X**. See the opportunities for *n* in the table above. |
| **XEQ** | XEQ *label* | PRG | Calls the respective subroutine. |
| | | \PRG | Executes the respective program. |
| | B , C , or D | PRG | Calls the respective subroutine if defined. |
| | | \PRG, \a | Executes the respective program if defined. |
| | f B , C , or D | a | |
| XNOR | h X.FCN **XN**OR | b, 8, d, h | |
| **XOR** | h XOR | b, 8, d, h | |
| ***x*** $^2$ | g $x^2$ | \a | |
| ***x !*** | h ! | DECM | |
| x → α | h X.FCN **X→α** | \a | Works like XTOA in HP-42S. |
| ***x*↔** | h x≷ *reg* | \a | See opportunities for *reg* in the table above.<br>CPX h x≷ will exchange ***x*** and ***y*** with the register specified and the next adjacent register. |
| ***x*↔*y*** | x≷y | \a | This performs **Re <> Im** if a complex operation was executed before.<br>CPX x≷y will exchange ***x*** and ***y*** with ***z*** and ***t***. |
| **x < … ?** | f x<? *arg* | \a | Compares ***x*** with *arg*. See opportunities for *arg* in the table above. The three dots will be replaced in the listing by *arg* according to the samples given in said table. |
| **x ≤ … ?** | g x≤? *arg* | | |
| **x = … ?** | f x=? *arg* | | |
| **x ≠ … ?** | g x≠? *arg* | | |
| **x ≥ … ?** | f x≥? *arg* | | |
| **x > … ?** | g x>? *arg* | | |
| Y.MD | h X.FCN **Y**.MD | DECM | Sets the format for date calculations. |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| ***y^x*** | g $y^x$ | DECM | |
| | C | DECM | As long as no reassignment took place. |
| **z(p)** | g zP | DECM | Like in $\mathbf{Q^{-1}}$ in HP-32E and $\mathbf{z_P}$ in HP-21S. |
| αAPP | h X.FCN f α **APP** *char* | \a | Switches to alpha mode for the input of 1 character, appends this to the alpha register, and returns to the mode set before. Eventually, this equals the sequence αON *char* αOFF. |
| αIP | h X.FCN f α **IP** | \a | Like AIP in HP-42S. |
| αLENG | h X.FCN f α **LENG** | \a | Like ALENG in HP-42S. |
| **αRCL** | f RCL *reg* | a | See opportunities for *reg* in the table above. |
| αROT | h X.FCN f α **ROT** | \a | Like AROT in HP-42S. |
| αSHIFT | h X.FCN f α **SHIFT** | All | Like ASHF in HP-42S. |
| **αSTO** | f STO *reg* | a | See opportunities for *reg* in the table above. |
| **αVIEW** | h X.FCN f α **VIEW** | \a | |
| α → x | h X.FCN f α →x | \a | Like ATOX in HP-42S. |
| ΔDAYS | h X.FCN **ΔDAYS** | DECM | Calculates the number of days between 2 dates ***x*** and ***y***. Function works like in 12C. |
| **Δ%** | g Δ% | DECM | |
| ***π*** | h π | DECM | CPX h π puts π in **X** and clears **Y** for using π in complex calculations. |
| | D | DECM | As long as no reassignment took place. |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| Σlnx | h X.FCN **ΣLNX** | DECM | Recalls the respective statistical sum. These sums are necessary for the curve fitting models beyond pure linear. See below for more. |
| $\Sigma ln^2x$ | h X.FCN **ΣLN2X** | | |
| Σlnxy | h X.FCN **ΣLNXY** | | |
| Σlny | h X.FCN **ΣLNY** | | |
| $\Sigma ln^2y$ | h X.FCN **ΣLN2Y** | | |
| Σxlny | h X.FCN **ΣXL**NY | | |
| Σylnx | h X.FCN **ΣYL**NX | | |
| 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. |
| $\Sigma x^2$ | h X.FCN **ΣX2** | | |
| Σxy | h X.FCN **ΣXY** | | |
| Σy | h X.FCN **ΣY** | | |
| $\Sigma y^2$ | h X.FCN **ΣY2** | | |
| **Σ+** | Σ+ | DECM | |
| **Σ–** | h Σ– | DECM | |
| 1COMPL | h X.FCN **1**COMPL | \a | Like 1's complement in HP-16C. |
| ***10^x*** | g $10^x$ | DECM | |
| ***1/x*** | f 1/x | DECM | |
| | B | DECM | As long as no reassignment took place. |
| 2COMPL | h X.FCN **2**COMPL | \a | Like 2's complement in HP-16C. |
| ***2^x*** | g $2^x$ | \a | |
| **+** | + | DECM | |
| **–** | – | | |
| **×** | × | | |
| **/** | / | | |
| **+/–** | +/– | | |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| // | g // | DECM | Calculates $\left(\frac{1}{x}+\frac{1}{y}\right)^{-1}$. |
| √‾ | f √x | \a | |
| % | f % | DECM | |
| %T | h X.FCN<br>h % T | DECM | Calculates $\frac{x}{y}\cdot 100$. |
| %+ | h %+ | DECM | Adds a markup of ***x*** % to ***y***. |
| %– | h %– | DECM | Subtracts a discount of ***x*** % from ***y***. |
| →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 | | |
| →H | → f H | H.MS | Takes the hours or degrees in **X** and converts them into decimal numbers. |
| →H.MS | → g H.MS | DECM | Assumes **X** containing *decimal* hours or degrees and converts them in the format HHH.MMSS. |
| →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. |
| →REC | f ▶R | DECM | Assumes **X** and **Y** containing the coordinates r and ϑ and converts them to x and y. |
| | | | **Pure input commands:** |
| **0, 1** | 0 , 1 | All | Numeric input. The 6 top left keys will be used for input of hexadecimal numbers >10 in HEXM by default. |
| **2 … 7** | 2 …7 | \b | |
| **8, 9** | 8 , 9 | \b, \8 | |
| **A … F** | A etc. | h | |
| **E** | E | 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. |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| **[ ]** or **[ / ]** | [,] | /c | First input inserts a space, second a fraction mark, e.g. [2] [,] [3] [,] [4] results in 2 ¾ in the display. |
| **[°]** | [,] | H.MS | Separates degrees (or hours) from minutes and seconds. |
| **[.]** | [,] | Y.MD, D.MY, M.DY | Separates the leading unit in date modes. It is left to the user to decide if a number displayed represents a date or not. |
| | | | **Catalogues (non programmable):** |
| ***CONST*** | [RCL] [C] | DECM | Calls the catalogue of constants like in HP35s (see table below). The constants stored are listed in a separate table below. [▲] and [▼] browse the catalogue, [ENTER↑] selects the constant displayed and recalls it into **X**. Use [EXIT] to leave the catalogue without picking a constant.<br>[CPX] [RCL] [C] will clear **Y** in recalling. |
| **CONV** | [h] [CONV] | DECM | Calls the catalogue of conversions (see table below). [▲] and [▼] browse the catalogue, [ENTER↑] converts ***x*** according to the conversion displayed. Use [EXIT] to leave the catalogue without converting. |
| ***X.FCN*** | [h] [X.FCN] | DECM | Calls the catalogue of extra real functions and sets alpha mode to allow for keying in names of operations. [▲] and [▼] browse the catalogue, [ENTER↑] selects the function displayed and executes it. You may also use [EXIT] to leave the catalogue without picking a function.<br>[CPX] [h] [X.FCN] calls the catalogue of extra complex functions. |
| | | b, 8, d, h | Calls the catalogue of extra integer functions. See above for more. |

| Name in listings | Keys to press | Works in modes | Remarks |
|---|---|---|---|
| | | | **Pure navigation and information commands (non programmable):** |
| | ▲ | Catalogue open | Go to previous item. |
| | | PRG | Go 1 step back in program memory. |
| | | \PRG, \a | Go 1 step back without executing this step. |
| | ▼ | Catalogue open | Go to next item. |
| | | PRG | Go 1 step forward in program memory. |
| | | \PRG, \a | Go 1 step forward executing this step. |
| | EXIT | All | Exits catalogues and any other operations with pending input, canceling the execution of this operation. |
| | g OFF | All | |
| | ON | Calc. off | |
| | h PRGM | \a | Toggles programming mode. |
| | h SHOW | DECM | Shows the full mantissa. |
| | h STATUS | \PRG, \a | Similar to HP-16C. |
| | f ◀\| | b, \PRG | Shift the display window like in HP-16C. |
| | g \|▶ | | |
| αOFF | ENTER↑ | a | Toggles alpha mode like AOFF and AON in HP-42S.<br>*) Please see the table for register addressing above for details. |
| αON | f α | \a | |
| | ENTER↑ | *) | |
| →BIN | → f 2 | \a | Shows ***x*** in target representation until the next command is executed. Mode is kept constant. |
| →DEC | → f 10 | | |
| →HEX | → g 16 | | |
| →OCT | → g 8 | | |

## TABLE OF CONSTANTS

This lists all constants contained in the catalogue CONST. The constants **π** and **e** are found on the keyboard directly. Names printed bold represent fundamental or measured constants, while the other ones may be derived from them.

| Name | Value | Dimension | Remarks |
|---|---|---|---|
| a | 365,2425 | d | Gregorian year (per definition) |
| $a_0$ | 5,291772083E-11 | m | Bohr radius $= \alpha/4\pi R_\infty$ |
| **atm** | 101325 | Pa / atm | Standard pressure $p_0$ |
| **c** | 299792458 | m / s | Vacuum speed of light (per definition) |
| $c_1$ | 374177107E-16 | W m² | First radiation constant $= 2\pi \cdot h \cdot c^2$ |
| $c_2$ | 0,014387752 | m K | Second radiation constant $= hc/k$ |
| **eV** | 1,602176462E-19 | J = A s V | = Electron charge x 1V |
| F | 96485,3415 | A s / mol | Faraday's constant = e $N_A$ |
| **g** | 9,80665 | $m/s^2$ | Standard earth acceleration |
| **G** | 6,6742867E-11 | $\frac{m^3}{kg \cdot s^2}$ | Newton's gravitation constant |
| $\mathbf{g_e}$ | 2,002319304362 | 1 | Landé g-factor |
| $G_o$ | 7,748091696E-5 | 1 / Ω | Conductance quantum $= 2e^2/h$ |
| **h** | 6,62606876E-34 | J s | Planck constant |
| $\hbar$ | 1,054571596E-34 | J s | $= h/2\pi$ |
| k | 1,3806503E-23 | J/K | Boltzmann constant $= R/N_A$ |
| $\mathbf{m_e}$ | 9,10938188E-31 | kg | Electron mass |
| $\mathbf{m_n}$ | 1,67492716E-27 | kg | Neutron mass |
| $\mathbf{m_p}$ | 1,67262158E-27 | kg | Proton mass |
| $m_u$ | 1,66053873E-27 | kg | Atomic unit mass = 1g / $N_A$ |
| $\mathbf{m_\mu}$ | 1,88353109E-28 | kg | Myon mass |
| $\mathbf{N_A}$ | 6,02214199E23 | 1/mol | Avogadro's number |

| Name | Value | Dimension | Remarks |
|---|---|---|---|
| $r_e$ | 2,817940285E-15 | m | Classical electron radius $= \alpha^2 \cdot a_0$ |
| **R** | 8,314472 | $\frac{J}{mol \cdot K}$ | Molar gas constant |
| $R_K$ | 25812,80756 | Ω | Von Klitzing constant = *h / e²* |
| $R_\infty$ | 10973731,5685 | 1 / m | Rydberg constant $= \alpha^2 m_e c / 2h$ |
| **$T_o$** | 273,15 | K | = 0°C, standard temperature (per definition) |
| $V_m$ | 0,022413996 | $m^3/mol$ | Molar volume of ideal gas at standard conditions $= RT_0 / p_0$ |
| $Z_o$ | 376,730313461 | Ω | Characteristic impedance of vacuum $= \sqrt{\mu_0 / \varepsilon_0} = \mu_0 c$ |
| $\alpha$ | 7,297352533E-3 | 1 | Fine-structure constant $= e^2 / 4\pi\varepsilon_0 \hbar c$ |
| **$\gamma_{EM}$** | 0,5772156649015328606 | 1 | Euler-Mascheroni's constant |
| $\gamma_p$ | 267522212 | $\frac{1}{s \cdot T}$ | Proton gyromagnetic ratio $= 2\mu_P / \hbar$ |
| $\varepsilon_o$ | 8,854187817E-12 | $\frac{A \cdot s}{V \cdot m}$ | Electric constant, vacuum permittivity $= 1 / \mu_0 c^2$ |
| $\lambda_c$ | 2,426310215E-12 | m | Compton wavelength $= h / m_e c$ |
| $\lambda_{cn}$ | 1,319590898E-15 | m | Compton wavelength of neutron $= h / m_n c$ |
| $\lambda_{cp}$ | 1,321409847E-15 | m | Compton wavelength of proton $= h / m_p c$ |
| $\mu_B$ | 9,27400899E-24 | J / T | Bohr's magneton $= e\,\hbar / (2\,m_e)$ |
| **$\mu_e$** | -9,28476362E-24 | J / T | Electron magnetic moment |
| $\mu_N$ | 5,05078317E-27 | J / T | Nuclear magneton $= e\,\hbar / (2\,m_p)$ |
| **$\mu_n$** | -9,662364E-27 | J / T | Neutron magnetic moment |
| **$\mu_p$** | 1,410606633E-26 | J / T | Proton magnetic moment |
| $\mu_o$ | 1,2566370614E-6 | $N / A^2$ | Magnetic constant, vacuum permeability $= 4\pi \cdot 10^{-7}$ (per definition) |

| Name | Value | Dimension | Remarks |
|---|---|---|---|
| $\mu_\mu$ | -4,49044813E-26 | J / T | Myon magnetic moment |
| $\sigma$ | 5,6704E-8 | $\frac{W}{m^2 \cdot K^4}$ | Stefan Boltzmann constant<br>= $2\pi^5 k^4 / (15 h^3 c^2)$ |
| $\Phi$ | 1,6180339887498948482 | 1 | Golden ratio $\frac{1+\sqrt{5}}{2}$ |
| $\Phi_o$ | 2,067833636E-15 | V s | Magnetic flux quantum $= h/2e$ |
| $\infty$ | | 1 | Infinity |

## TABLE OF CONVERSIONS

This lists all conversions contained in the new catalogue CONV. The constants **atm** and $\mathbf{T_o}$ may be useful for conversions, too – they are found in the catalogue CONST.

| Conversion | Remarks |
|---|---|
| acres→ha | Remember 1 ha = $10^4$ m² |
| au→km | Astronomic units |
| cm→inch | |
| feet→m | |
| flozUK→ml | Remember 1 m³ = $10^3$ l |
| flozUS→ml | |
| g→oz | |
| galUK→l | |
| galUS→l | |
| ha→acres | |
| inch→cm | |
| kg→lbm | |
| km→au | |
| km→ly | |
| km→mi | |
| km→nmi | Nautical miles |
| l→galUK | |
| l→galUS | |
| lbm→kg | |
| ly→km | Light years |

| Conversion | Remarks |
|---|---|
| m→feet | |
| mi→km | |
| ml→flozUK | |
| ml→flozUS | |
| nmi→km | |
| oz→g | |
| °C→°F | |
| °F→°C | |

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

1. **TIMER** (accuracy down to 0.1s is sufficient, we can't press keys more precisely – included in the index already but not implemented yet)
2. **T**, **CHISQ**, and **F** distributions (in this order) and their inverses (not in the index yet)

| 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. |
| 1.4 | 10.2.09 | Added CONST and a table of constants provided, D>J and J>D, LEAP?, %T, RCL and STO ▲ and ▼, and 2 forgotten statistics registers; deleted CHS, EMGAM, GSB, REAL and ZETA; purged and renamed the bit operations; renamed many commands. |
| 1.5 | 5.3.09 | Added RNDINT, CONV and its table, a memory table, the description of XEQ B, C, D to the operation index, and $a$ and $g_e$ to the table of constants; put CLSTK on a key, moved CLΣ and FILL, changed the % and log labels on the keyboard, put CLALL in X.FCN; checked and cleaned alpha mode keyboard and added a temporary alpha keyboard; rearranged the alphabet to put Greek after Latin, symbols after Greek consistently; separated the input and non programmable commands; cleaned the addressing tables. |