

# HP-67

## Quick Reference Card

### AUTOMATIC MEMORY STACK

T	0.00
Z	0.00
Y	0.00
X	0.00

Always displayed.

### STORAGE REGISTERS

#### Primary Registers

(n) Address

I	25
R <sub>E</sub>	24
R <sub>D</sub>	23
R <sub>C</sub>	22
R <sub>B</sub>	21
R <sub>A</sub>	20

#### Secondary Registers

(n) Address

R <sub>9</sub>	9	R <sub>S9</sub>	n	19
R <sub>8</sub>	8	R <sub>S8</sub>	$\Sigma xy$	18
R <sub>7</sub>	7	R <sub>S7</sub>	$\Sigma y^2$	17
R <sub>6</sub>	6	R <sub>S6</sub>	$\Sigma y$	16
R <sub>5</sub>	5	R <sub>S5</sub>	$\Sigma x^2$	15
R <sub>4</sub>	4	R <sub>S4</sub>	$\Sigma x$	14
R <sub>3</sub>	3	R <sub>S3</sub>		13
R <sub>2</sub>	2	R <sub>S2</sub>		12
R <sub>1</sub>	1	R <sub>S1</sub>		11
R <sub>0</sub>	0	R <sub>S0</sub>		10

**STO** (n) stores x value in R<sub>n</sub>.

**RCL** (n) recalls value from R<sub>n</sub>.

**h** **STI** stores x value in I-register.

**h** **RCL** recalls value from I-register.

**f** **P $\pm$ S** exchanges contents of primary registers with secondary registers. Used to access secondary storage registers for **STO** and **RCL**.

**STO** + (n), **STO** - (n), **STO**  $\times$  (n), **STO**  $\div$  (n); x value added to, subtracted from, multiplied by, or divided into contents of R<sub>n</sub>, and result placed in R<sub>n</sub>. Can be performed on primary storage registers R<sub>0</sub> through R<sub>9</sub>.

**f** **CLREG** clears all primary storage registers to 0.00

### ACCUMULATIONS

Press **f** **CLREG** followed by **f** **P $\pm$ S** to clear secondary storage registers R<sub>S0</sub> through R<sub>S9</sub> before using  **$\Sigma^+$** .

**$\Sigma^+$**  stores accumulations of number in the X- and Y-registers of the stack in secondary storage registers R<sub>S4</sub> through R<sub>S9</sub> as shown in the illustration of the storage registers shown above.

### FLAGS

F0 set by **h** **SF** 0; cleared by **h** **CF** 0.

F1 set by **h** **SF** 1; cleared by **h** **CF** 1.

F2 set by **h** **SF** 2; cleared by **h** **CF** 2 or when tested by **h** **F?** 2.

F3 set by **h** **SF** 3 or by digit entry from keyboard or by loading data from a magnetic data card; cleared by **h** **CF** 3 or when tested by **h** **F?** 3.

Test-cleared flags F2 and F3 are tested first by **F?**, then cleared.

### USING THE I-REGISTER FOR INDIRECT CONTROL

A value can be placed in the I-register using either the **h** **STI** or the **h** **x $\pm$ 1** operations. The (n) function then uses

the absolute value of the integer portion of the number stored in I as an address or control value. If the number in I is outside the specified limits, an error condition occurs. Indirect control can be performed with:

**STO (i)**, **RCL (i)**; with 0 through 25 in I, stores x value in or recalls x-value from primary or secondary storage register addressed by number in I.

**STO + (i)**, **STO - (i)**, **STO × (i)**, **STO ÷ (i)**; with 0 through 25 in I, x-value added to, subtracted from, multiplied by, or divided into contents of storage register addressed by number in I-register; and result placed in storage register addressed by number in I.

**DSP (i)**, with number in I-register from 0 through 9, specifies **DSP** 0 through **DSP** 9.

**g** **ISZ (i)**, **g** **DSZ (i)**, with number in I-register from 0 through 25, increments (adds 1 to) or decrements (subtracts 1 from) storage register addressed by number in I-register. In a running program, if value in addressed storage register is then 0, execution skips one step of program memory before continuing.

**GTO (i)**, **f** **GSB (i)**, with number in I-register from 0 through 19, transfers execution to portion of program defined by the selected label, according to the address scheme below:

<b>(i)</b> Address	<b>(i)</b> Address
<b>f</b> <b>LBL</b> 0      0	<b>f</b> <b>LBL</b> <b>A</b> 10
<b>f</b> <b>LBL</b> 1      1	<b>f</b> <b>LBL</b> <b>B</b> 11
<b>f</b> <b>LBL</b> 2      2	<b>f</b> <b>LBL</b> <b>C</b> 12
<b>f</b> <b>LBL</b> 3      3	<b>f</b> <b>LBL</b> <b>D</b> 13
<b>f</b> <b>LBL</b> 4      4	<b>f</b> <b>LBL</b> <b>E</b> 14
<b>f</b> <b>LBL</b> 5      5	<b>g</b> <b>LBL</b> <b>f</b> 15
<b>f</b> <b>LBL</b> 6      6	<b>g</b> <b>LBL</b> <b>b</b> 16
<b>f</b> <b>LBL</b> 7      7	<b>g</b> <b>LBL</b> <b>c</b> 17
<b>f</b> <b>LBL</b> 8      8	<b>g</b> <b>LBL</b> <b>d</b> 18
<b>f</b> <b>LBL</b> 9      9	<b>g</b> <b>LBL</b> <b>e</b> 19

With -1 through -999 in the I-register, **GTO (i)** or **f** **GSB (i)** transfers execution *back* in program memory the number of steps specified by the number in I.

## MAGNETIC CARD OPERATIONS—DATA

### Recording Data

To record data from the storage registers in the calculator onto a magnetic card, set the W/PRGM-RUN switch to RUN, press **f** **W/DATA**, and pass an unprotected magnetic card through the card reader. If the calculator display shows **Crd** , reverse the card and pass the second side through.

### Loading Data

To load data from a magnetic card into the calculator's storage registers, set the W/PRGM-RUN switch to RUN and pass a side of the magnetic card containing data through the card reader. If the calculator displays **Crd** , reverse the card and pass the second side through.

To load *some* data from a magnetic card into the calculator's storage registers while preserving the contents of other calculator registers, set the W/PRGM-RUN switch to RUN and store in the I-register the number address of the last storage register to be loaded from the card. Then press **g** **MERGE** and pass one side of the magnetic card through the card reader. If the calculator displays **Crd** , reverse the card and pass the second side through. Data will be loaded from the card beginning with register  $R_0$  up to and including the storage register specified by the number in I. The contents of the remaining storage registers in the calculator remain unchanged.

# PROGRAMMING THE HP-67

## PROGRAM MEMORY

000	
001	84
002	84
003	84

Top-of-memory marker.

223	84
224	84

224 steps for your program instructions.

When the HP-67 is first turned ON, program memory is filled with **R/S** instructions (keycode 84).

## PROGRAM MODE WIPRGM RUN

In program mode, only the following five functions are active. All other functions are loaded into program memory when pressed. Passing a magnetic card through the card reader records the contents of program memory on the card.

**GTO**     sets calculator to step *nnn* of program memory.

**SST** Single step. Moves calculator forward one step of program memory.

**h** **BST** Back step. Moves calculator back one step in program memory.

**h** **DEL** Delete. Deletes current instruction from program memory. All subsequent instructions moved up one step.

**f** **CLPRGM** Clear program. Clears program memory to all **R/S** instructions, sets calculator to step 000, clears all flags, and specifies **FIX 2** display and **DEG** trigonometric modes.

## RUN MODE WIPRGM RUN

**GTO**     sets calculator to step *nnn* of program memory.

Some functions that are active in **PROGRAM** mode operate differently in **RUN** mode:

**SST** Single step. Displays step number and keycode of current program memory step when held down; executes current instruction, displays result, and moves to next step when released. Used for single-step execution of program.

**h** **BST** Back step. Moves to previous step and displays step number and keycode of that previous memory step when **BST** is held down; displays original contents of X-register when released. No instructions are executed.

**h** **DEL** does nothing except cancel the **h** prefix key.

**f** **CLPRGM** does nothing except cancel the **f** prefix key.

## Executed in a Program

Function keys may be executed as instructions in a program. Some function keys which are most often used in or are unique to programming applications are shown below:

**A** **B** **C** **D** **E** Label designators. When preceded by **LBL**, **0** **1** **2** **3** **4** define beginning of routine. When preceded by **GTO** or **GSB**, cause calculator to stop execution, search downward through program memory to first designated label, and resume execution there.

**a** **b** **c** **d** **e** Label designators. Operate exactly as label designators listed above, except they are preceded only by **LBL**, **GTO**, **f**, and **GSB**.

**PAUSE** Stops program execution and transfers control to keyboard for 1 second, then resumes program execution.

**X≠Y** **X<Y** **X>Y** **X≤Y** Conditionals. Each tests value in **X**-register against 0 or value in **Y**-register as indicated. If true, calculator executes instruction in next step of program memory. If false, calculator skips one step before resuming execution.

**F2** If flag true. Followed by flag designator (0, 1, 2, or 3), tests designated flag. If flag is set (true) the calculator executes the instruction in the next step of program memory. If flag is cleared (false), calculator skips one step before resuming execution. **F2** clears flags F2 and F3 after test.

**-X-** Flash X. Pauses to display contents of X-register for 5 seconds. Used to write down answers or to interface programs with HP-97 Programmable Calculator printer.

**SPACE** Executed as no operation in HP-67. Used to interface programs with HP-97 Programmable Calculator printer.

**RTN** Return. Executed as result of pressing a label designator or executing a **GTO** instruction, stops execution and returns control to keyboard. Executed as a result of a **GSB** instruction, returns control to next step after the **GSB** instruction.

**R/S** Run/stop. Stops program execution.

### Pressed from the Keyboard

$\frac{1}{x}$   $\sqrt{x}$   $y^x$   $R\pm$   $x\div y$  Default functions. Operate only in RUN mode when no instructions have been loaded into program memory. Duplicated by other functions on calculator for programming or manual use.

**A B C D E** User-definable keys. Cause calculator to **a b c d e** search downward through program memory to first designated label and begin execution there.

**GTO** Go to. Followed by label designator (**A** through **E**, **f a** through **f e**, **0** through **9**, **(**), **)**), causes calculator to search downward through program memory to first designated label and stop there.

**GSB**, **GSB1** Go to subroutine. Followed by label designator, (**A** through **E**, **0** through **9**, **(**), **)**, or **a** through **e**), cause calculator to search downward through program memory to first designated label and execute that section of program memory as a subroutine. Subroutines can be nested up to

three levels deep.

**RTN** Return. Sets calculator to step 000 of program memory.

**R/S** Run/stop. Begins execution from current step of program memory. Stops execution if program is running.

Any key. Pressing any key on the keyboard stops execution of a running program.

## MAGNETIC CARD OPERATIONS—PROGRAMS

### Recording a Program

To record a program from the calculator onto a magnetic card, set the W/PRGM-RUN switch to W/PRGM and pass one side of the card through the card reader. If the calculator display shows **Crd**  , reverse the card and pass the second side through.

### Loading a Program

To load a program from a magnetic card into the calculator, set the W/PRGM-RUN switch to RUN and pass one side of the card through the card reader. If the calculator displays **Crd**  , pass the second side through.

To preserve a portion of a program in the calculator while merging a second program from a card after it, set the W/PRGM-RUN switch to RUN and set the calculator (e.g., with **GTO** **0** **n n n**) to the last step of program memory which you want to preserve. Then press **g** **MERGE** and pass one side of the magnetic card containing the new program through the card reader. If the calculator displays **Crd**  , reverse the card and pass the second side through. Now instructions from the card have been loaded into the calculator beginning with the step of program memory after the step to which the calculator is set.

HEWLETT  PACKARD