An HP16C/WP 34S Dictionary

Jake Schwartz

The RPN-based HP16C, introduced on July 1, 1982 (along with the HP15C on the same day) represented Hewlett-Packard's first attempt to produce a "programmer's calculator", with an assortment of functions for integers in binary, octal, decimal or hexadecimal bases. Included were (among others) integer arithmetic in unsigned, 1's-complement or 2's complement modes; bit shifting, rotating, masking and logical (AND, OR, NOT, etc.) operators; as well as accounting for potential carry and overflow conditions. The word size could be set to anything between 1 and 64 bits and the calculator would respect this word size in all its operations, including adjusting its 203 bytes of RAM by expanding or contracting the number of accessible data registers depending on word size.

Machines (such as the HP42S, 48 /49/50, 35S) from follow-on HP calculator generations accommodated some of the HP16C's functionality, but no unit completely ported the full function set. Having been frustrated with this situation, Rick Grevelle and I created the HP16C Emulator Library in 1993, which brought the entire 16C capability to the HP48SX (via a re-definable Mylar keyboard overlay). Following this, nothing came close until the advent of the WP 34S, the recent brain-child of Walter Bonin, Paul Dale and Marcus Von Cube. Repurposed from an HP20b or HP30b, this function set not only includes virtually the entire HP16C functionality, but far exceeds it in many ways. With the machine's modern CPU, more updated LCD and larger memory, the 34S permits 12 digits of integers to be displayed (versus 8 on the 16C), expands program memory to *multiple banks* of 506 steps (versus a maximum of 203 steps at the expense of data registers for the 16C), displays program steps in alphanumeric format (as opposed to 16C key codes) and runs at least 100 times as fast. This is not to mention the 112 data registers, 104 flags, ALPHA register and ability to run with either a 4- or 8-level RPN stack.

Figure 1. (L to R) HP16C, HP48SX keyboard with HP16C Emulator Library overlay attached, WP 34S.

First, table 1 shows a brief overview of the comparison:

HP16C		WP 34S
ENTER, R↓, R↑ X<>Y, CLx, LASTx, FLOAT Mode, Leading zeros on/off (SF3, CF3)	**Stack & Display Manipulation**	ENTER, R↓, R↑ X<>Y, CLx, RCL L, H .d, LZON, LZOFF, CLSTK, DROP, FILL
Arithmetic, √, 1/x, ABS, RMD, Double-word multiply, divide and remainder	**Math**	Arithmetic, √, 1/x, ABS, RMDR, Double-word multiply, divide, and remainder, x^2, $\log_2$, y^x
HEX, DEC, OCT, BIN	**Base Conversions**	HEX, DEC, OCT, BIN and any integer base between 2 and 16
1's, 2's, UNSGN, WSIZE 1 – 64, 8-digit WINDOW chunks on displayed number	**Complement Modes and Word Size**	1COMPL, 2COMPL, UNSIGN, WSIZE 1 – 64, 12-digit chunks left-shift, right-shift of displayed number
Flags 0 – 5, SF, CF, F?	**Flag Manipulation**	Flags 0-99 & A-D, SF, CF, FF, FS?, FS?C, FS?S, FS?F, CLFLAG
SB, CB, B?, #B	**Bit Manipulation**	SB, CB, FB, BS?, BC?, nBITS
AND, OR, NOT, XOR	**Logical Operations**	AND, OR, NOT, XOR, NAND, NOR, XNOR
RL, RR, RLn, RRn, RLC, RRC, RLCn, RRCn, SL, SR, ASR, LJ, MASKL, MASKR	**Rotate/Shift/Mask Operations**	RL 01, RR 01, RL n, RR n, RLC 01, RRC 01, RLC n, RRC n, SL 01, SR 01, SL n, SR n, ASR 01, LJ, RJ, MASKL n, MASKR n
STO, RCL, STO I, RCL I, STO (i), RCL (i), X<>I, X<>(i), CLEAR REG	**Register Access Operations**	STO n, RCL n, STO → n, RCL → n, X<> n, X<> → n, storage register arithmetic, recall register arithmetic, CLREG

Table 1. HP16C and WP 34S high-level function comparison.

For those of us who have always considered the HP16C to be essential, it seemed logical to create a concise dictionary to "translate" HP16C functions to WP 34S ones, showing the comparable keystrokes which achieve the same outcome. The result is the four-page table 2 starting on the next page. The table columns consist of the HP16C keystroke sequence, the function description, the WP 34S keystroke sequence, and finally any additional 34S functions which are related. In cases where the 34S performs functions not in the 16C's bag of tricks, those items are listed on separate lines. Hopefully, this will be of use to those who have discovered the 34S, can't put it down and wish to carry around one less calculator.

16C Function	Description	34S Function	Alternate 34S Function(s)
ON	Turn on....	ON	
ON	Turn off	g OFF	
ON / -	Reset continuous memory		
ON / ·	Change digit separator	h ./,	
g BSP	Clears last digit or whole disp	←	
g CLx	Clear X	h CLx	
f CLEAR PRGM	Clear program memory	h CLP	
f CLEAR REG	Clear registers	h X.FCN CLREG	
f CLEAR PREFIX	Cancel prefix entry	(Press shift key again)	
0 to 9, A to F	Digit entry	0 … 9 A … F	
·	Decimal point	.	
ENTER	Duplicate X into Y	ENTER↑	
CHS	Change sign	+/-	
f EEX	Enter exponent	EEX	
x⇄y	X exchange Y	x⇄y	h x⇄ *r*
R↓, g R↑	Roll down, roll up	R↓ h R↑	
HEX DEC OCT BIN	Number base modes	f 2 g 8 f 10 g 16	
	Set other bases 3-15	h MODE BASE *n*	
f SHOW DEC etc.	Temp display X in specified base	(Recently removed)	
f SET COMPL 1's	1s compliment mode	h MODE 1COMPL	
f SET COMPL 2's	2s compliment mode	h MODE 2COMPL	
f SET COMPL UNSGN	Unsigned mode	h MODE UNSIGN	
f WSIZE	Set word size	h MODE WSIZE → X	h MODE WSIZE *n*
f WINDOW n	Displays specified 8-digit segment of the integer in X	f ◀ g ▶	
g < g >	Scroll left, scroll right by one digit		
g SF g CF	Set flag, clear flag	h P.FCN SF *n* h P.FCN CF *n*	h P.FCN FF *n*

16C Function	Description	34S Function	Alternate 34S Function(s)
f STATUS	Temp display current modes	h STATUS	
f FLOAT n or .	Floating-point decimal mode	f H.d	
+ − × ÷	Arithmetic Operators	+ − × /	
f RMD	Remainder	h RMDR	
g √x	Square Root	f √x	g x²
g 1/x	Reciprocal	f 1/x	
g DBLX	Double Multiply	h X.FCN DBL ×	
g DBL÷	Double Divide	h X.FCN DBL /	
g DBLR	Double Remainder	h X.FCN DBLR	
g ABS	Absolute Value	f |x|	
f SL f SR	Shift Left, Shift Right	h X.FCN SL 0 1 h X.FCN SR 0 1	h X.FCN SL *n* h X.FCN SR *n*
g ASR	Arithmetic Shift Right	h X.FCN ASR 0 1	h X.FCN ASR *n*
f RL f RR	Rotate Left, Rotate Right	h X.FCN RL 0 1 h X.FCN RR 0 1	
g RLC g RRC	Rotate Left through Carry, Rotate Right through Carry	h X.FCN RLC 0 1 h X.FCN RRC 0 1	h X.FCN RLC → Y h X.FCN RRC → Y
f RLn f RRn	Rotate Left n Bits, Rotate Right n Bits	h X.FCN RL *n* h X.FCN RR *n*	h X.FCN RL → Y h X.FCN RR → Y
g RLCn g RRCn	Rotate Left (or Right) through Carry n Bits	h X.FCN RLC *n* h X.FCN RRC *n*	
g LJ	Left Justify	h X.FCN LJ	h X.FCN RJ
f MASKL	Mask Left	h X.FCN MASKL *n*	h X.FCN MASKL → Y
f MASKR	Mask Right	h X.FCN MASKR *n*	h X.FCN MASKR → Y
f SB	Set Bit	h X.FCN SB *n*	h X.FCN SB → Y
f CB	Clear Bit	h X.FCN CB *n*	h X.FCN CB → Y
	Flip Bit (Invert Bit)	h X.FCN FB *n*	h X.FCN FB → Y
g #B	Number of Bits	h X.FCN nBITS	
f NOT f OR	NOT, OR	h NOT h OR	h X.FCN NOR

16C Function	Description	34S Function	Alternate 34S Function(s)
f AND f XOR	AND, XOR	h AND h XOR	h X.FCN NAND h X.FCN XNOR
STO RCL	Store, Recall	STO RCL	
STO I STO (i)	Store into Indirect Register, Store Indirectly	STO *n* STO → *n*	
RCL I RCL (i)	Recall into Indirect Register, Recall Indirectly	RCL *n* RCL → *n*	
	Storage register arithmetic	STO + *d* STO − *d* STO × *d* STO / *d*	STO ▲ *d* STO ▼ *d*
	Recall register Arithmetic	RCL + *s* RCL − *s* RCL × *s* RCL / *s*	RCL ▲ *s* RCL ▼ *s*
f x⇄I , f x⇄(i)	X interchange I, X interchange indirectly	h x⇄ *r* h x⇄ → *n*	
g LSTx	Last X	RCL L	
f MEM	Memory Available	Num. prog steps shown in program mode	
f CLEAR REG	Clear Registers	h X.FCN CLREG	
f CLEAR PRGM	Clear Program	h CLP	
GTO (i) GSB (i)	GoTo Indirectly, GoSub Indirectly	h GTO → *n* XEQ → *n*	
g DSZ g ISZ	Decrement and Skip if Zero, Increment and Skip if Zero	h P.FCN DSZ *r* h P.FCN ISZ *r*	f DSE *r* g ISG *r*
g P/R	Program / Run	h P/R	
g LBL	Label	f LBL *label*	
g RTN	Return	g RTN	
GSB	GoSub	XEQ *label*	
R/S	Run / Stop	R/S	
g PSE	Pause	h PSE *nn*	
GTO	GoTo	h GTO *label*	
GTO · *nnn*	GoTo line number	h GTO . *nnn*	
SST g BST	Single Step, Back Step	▼ ▲	
g F?	Flag Set?	h TEST FS? *n*	h TEST FS?C *n* h TEST FS?S *n* h TEST FS?F *n*

16C Function	Description	34S Function	Alternate 34S Function(s)
	Flag Clear?	h TEST FC? *n*	h TEST FC?C *n* h TEST FC?S *n* h TEST FC?F *n*
f B?	Bit Set?	h TEST BS? *n*	h TEST BC? *n*
g x=y g x≠y	X equals Y? X Not equal to Y?	f x = ? Y g x ≠ ? Y	f x = ? Z f x = ? T etc.
g x≤y g x>y	X less than or equal to Y? X greater than Y?	h TEST x ≤ ? Y h TEST x > ? Y	h TEST x ≤ ? 1 h TEST x > ? 1 h TEST x ≤ ? 0
g x=0 g x≠0	X equals 0? X not equal to 0?	f x = ? 0 g x ≠ ? 0	f x = ? 1 g x ≠ ? 1 h TEST x=+0 ? h TEST x=–0?
g x<0 g x>0	X less than 0? X greater than 0?	h TEST x < ? 0 h TEST x > ? 0	h TEST x < ? 1 h TEST x > ? 1
	X less than / less than or equal to reg. value	h TEST x < ? *n* h TEST x ≤ ? *n*	h TEST x < ? → *n* etc.
	x greater or equal / greater than reg. value	h TEST x ≥ ? *n* h TEST x > ? *n*	h TEST x ≥ ? 0 h TEST x ≥ ? 1
g SF 3 g CF 3	Turn on or off leading zeroes	h MODE LZON h MODE LZOFF	
	Clear all flags	h P.FCN CLFLAG	
	Clear Stack	h P.FCN CLSTK	0 g FILL
	Drop	h X.FCN DROP	

Table 2. Comparison of HP16C and WP 34s functions, listed in the order of the Function Summary and Index, starting on page 120 of the HP16C Owner's Handbook.