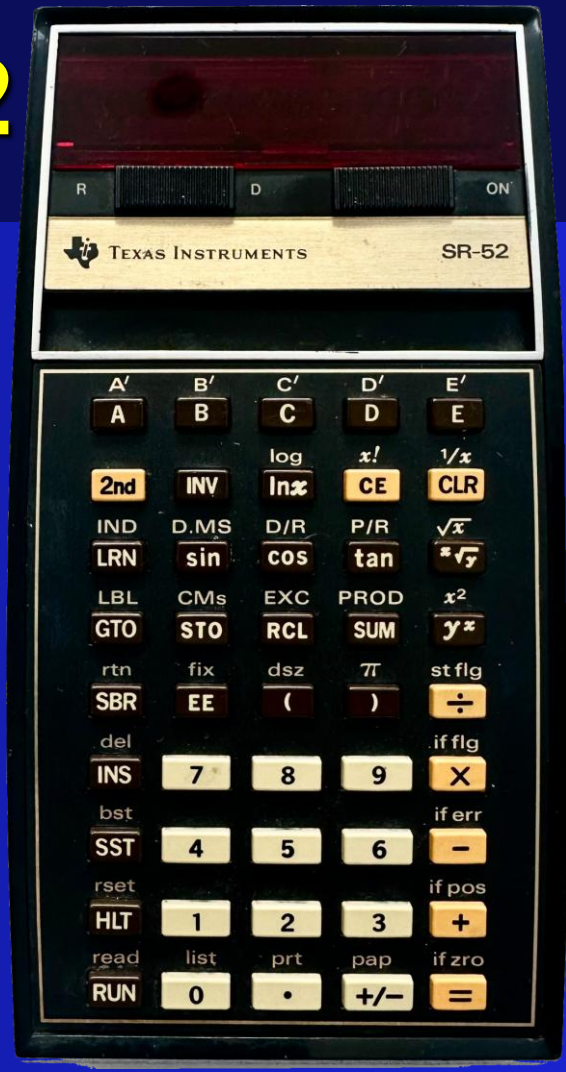
In praise of the SR-52





HHC 2026

October 31 – November 1, 2026

HHC 2026 – In praise of the SR-52

- The HP-65 had been introduced at $795 on January 19, 1974, although many of the early units had been made earlier – serial numbers 1333A.
- HP had the high-end, portable, programmable market to itself until ...
- TI introduced the SR-52 on September 16, 1975 at an SRP of $395. 18 months later.
- If the HP-65 is Mark 1, IMO the SR-52 is Mark 1.5 and the HP-67/97 is Mark 2. (my invented terms)

HHC 2026 – In praise of the SR-52

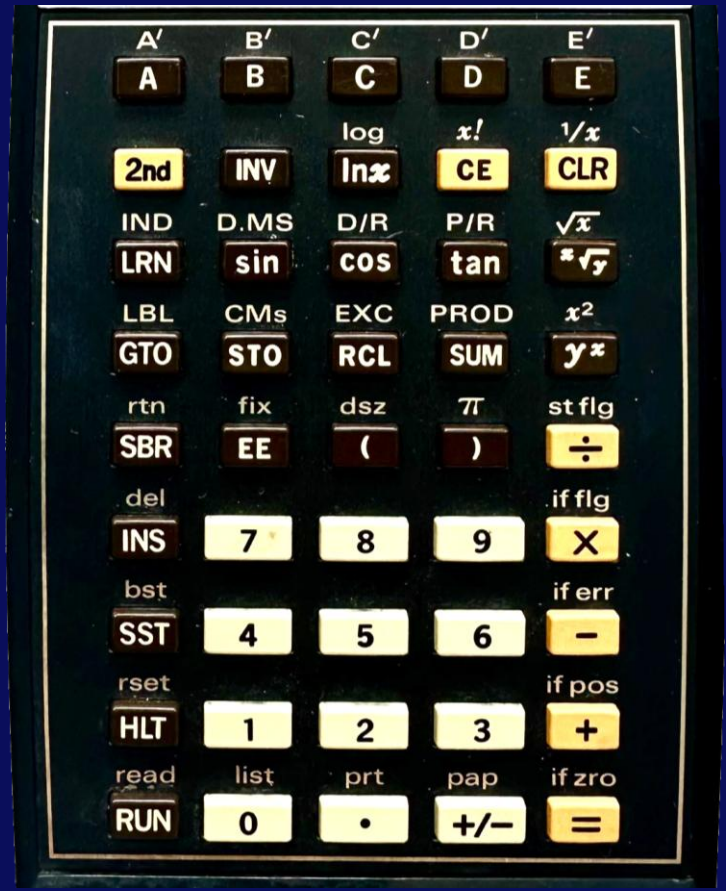
- The SR-52 continued TI's algebraic approach from the SR-50 / 51 but now used AOS.
- The Algebraic Operating System. Stop hissing!
- The SR-52 allowed up to 10 pending operations and 9 open sets of parentheses.
- This greatly increased the SR-52's capabilities.
- It is actually fairly difficult to exceed for any real problems. Even this one works keyed in as "written":

$$1+\left(2\times\left(3-\left(4\times\left(5+\left(\frac{6}{\left(7+\left(8^{(9+10^{11})}\right)\right)}\right)\right)\right)\right)\right)$$

HHC 2026 – In praise of the SR-52

Know the competition!

- 10+2 display
- 224 unmerged steps (2nd)
- 20 data registers (00-19)
- Subroutines – 2 levels
- DSZ looping: =0 and ≠0
- Five flags
- Printer available



HHC 2026 – In praise of the SR-52

Know the competition!

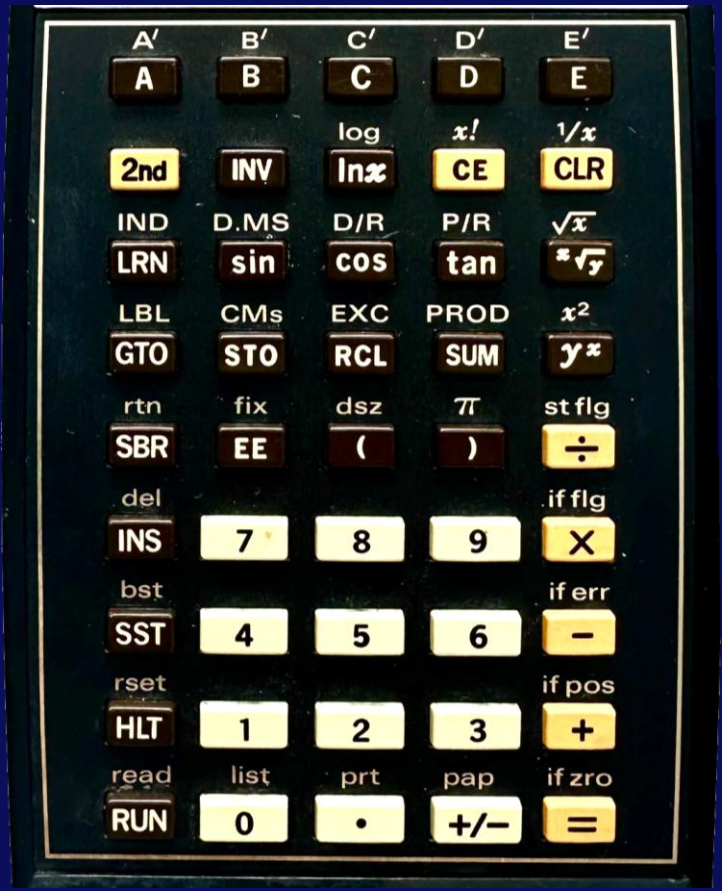
10 user defined keys

72 labels

Indirect capability

8 conditionals

- **FS? and FC? (If flg)**
- **Display =0 and ≠0 (if pos)**
- **Display >0 and <=0 (if zro)**
- **Error? and No error? (if err)**



HHC 2026 – In praise of the SR-52

Extra features – Memory registers

00 – 19: Base memories.

20 – 59: Void. RCL always returns 0.

Unless you added a memory chip on internal PCB.

60 – 69: AOS registers.

HHC 2026 – In praise of the SR-52

Extra features

- Calculations are fastest in FIX 8. Why?
- Memory registers 60 – 69: AOS registers.
  - "Fractured digits" – when storing / recalling from these registers, portions of the LED digit did not illuminate when recalled to display.
  - TI PPC Notes V2N2P5 "Yahtzee" display.
  - Used in games and to show DMS.

– In praise of the SR-52





HHC 2026 – In praise of the SR-52

Extra features – Memory registers

- 70 – 97: Program registers – 8 bytes per register.
  - These are valid arguments for STO and RCL.
  - Work well for data, not so much for program steps.
  - But allows entry of non-keyable keycodes.
- 98 – 99: Two free registers! Not affected by CMs.
  - Using memories 98-99, it is possible to write a program that will copy data memories into the program space which can then be recorded to and read from magnetic cards.

HHC 2026 – In praise of the SR-52

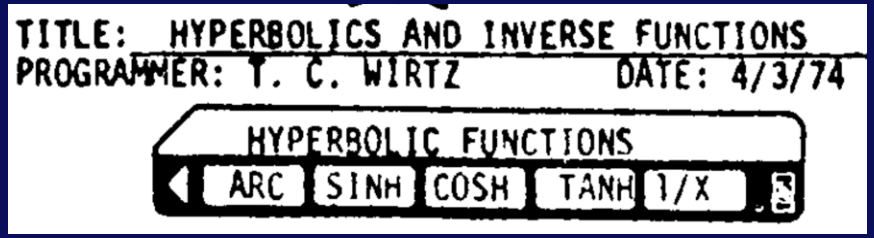
Extra features

- 12-digit calculations.
  - Numbers to / from the display truncated to 12 digits. Storage register arithmetic done to 13 places, so possible to get an extra digit's accuracy.
  - If Register 00 contains 3.141592653589 as a result of a calculation, recalling this to the display will lose the last digit. However, if you recall the register and see 3.1415926... You can key 3.1415926 INV SUM 00 to subtract this value from the register. Place the SR-52 into EE mode and RCL 00 to see the last digit.

HHC 2026 – SR-52 vs. HP-65 - Hyperbolics

HP-65. 65 Notes V1N5P5 – Ted Wirtz.

97/100 Steps (97%). 2/2 flags, 6/15 labels, 3/10 memories



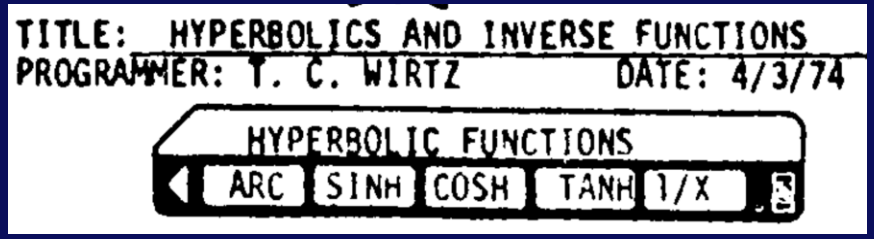
HP-65 HHC 2026 – SR-52 vs. HP-65 – Hyperbolics

f-1	E	RTN	GTO
SF1	+	LBL	0
f-1	g x<>y	A	LBL
SF2	g LSTx	RTN	D
R/S	-	LBL	C
LBL	f	C	1
E	TF1	f	g x<>y
g	g x<>y	SF1	+
1/x	g NOP	LBL	1
RTN	f-1	B	g LSTx
LBL	TF2	C	-
D	CLx	ENT	/
f	2	f-1	f
SF2	/	SQRT	LN
LBL	GTO	1	2
B	0	f	/
f	LBL	TF1	LBL
SF1	C	CHS	0
LBL	g RUP	g NOP	RCL 8
C	STO 8	+	RCL 7
C	g RUP	f	RCL 6
f-1	STO 7	SQRT	g RUP
LN	g RUP	+	g NOP
ENT	STO 6	f	g NOP
ENT	g RUP	LN	g NOP

HHC 2026 – SR-52 vs. HP-65 - Hyperbolics

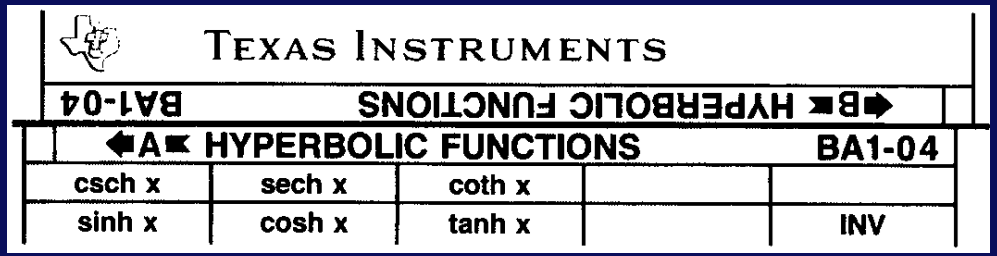
HP-65. 65 Notes V1N5P5 – Ted Wirth

97/100 Steps (97%). 2/2 flags, 6/15 labels, 3/10 memories.



SR-52. Basic Library BA1-4. (Factory software – Ugh)

191/224 Steps (85%). 2/5 flags, 18/72 labels, 2/20 memories



HP-65 HHC 2026 – HP-65 vs. SR-52 – Hyperbolics SR-52

f-1	E	RTN	GTO
SF1	+	LBL	0
f-1	g x<>y	A	LBL
SF2	g LSTx	RTN	D
R/S	-	LBL	C
LBL	f	C	1
E	TF1	f	g x<>y
g	g x<>y	SF1	+
1/x	g NOP	LBL	1
RTN	f-1	B	g LSTx
LBL	TF2	C	-
D	CLx	ENT	/
f	2	f-1	f
SF2	/	SQRT	LN
LBL	GTO	1	2
B	0	f	/
f	LBL	TF1	LBL
SF1	C	CHS	0
LBL	g RUP	g NOP	RCL 8
C	STO 8	+	RCL 7
C	g RUP	f	RCL 6
f-1	STO 7	SQRT	g RUP
LN	g RUP	+	g NOP
ENT	STO 6	f	g NOP
ENT	g RUP	LN	g NOP

PROGRAM LISTING

DISPLAY	KEY	DISPLAY	KEY	DISPLAY	KEY	DISPLAY	KEY
000 46	*LBL	028 77	*4'	056 50	*st flg	084 01	1
001 16	*A'	029 46	*LBL	057 02	2	085 56	*rtn
002 50	*st flg	030 18	*C'	058 56	*rtn	086 46	*LBL
003 02	2	031 50	*st flg	059 46	*LBL	087 89	*3'
004 46	*LBL	032 02	2	060 10	*E'	088 22	INV
005 11	A	033 46	*LBL	061 22	INV	089 60	*if flg
006 60	*if flg	034 13	C	062 23	Inx	090 02	2
007 01	1	035 60	*if flg	063 42	STO	091 79	*6'
008 87	*1'	036 01	1	064 01	1	092 20	*1/x
009 10	*E'	037 89	*3'	065 09	9	093 46	*LBL
010 02	2	038 10	*E'	066 42	STO	094 79	*6'
011 41	GTO	039 46	*LBL	067 01	1	095 42	STO
012 77	*4'	040 77	*4'	068 08	8	096 01	1
013 46	*LBL	041 22	INV	069 20	*1/x	097 09	9
014 17	*B'	042 49	*PROD	070 44	SUM	098 94	+/–
015 50	*st flg	043 01	1	071 01	1	099 42	STO
016 02	2	044 09	9	072 08	8	100 01	1
017 46	*LBL	045 43	RCL	073 94	+/–	101 08	8
018 12	B	046 01	1	074 44	SUM	102 01	1
019 60	*if flg	047 09	9	075 01	1	103 44	SUM
020 01	1	048 22	INV	076 09	9	104 01	1
021 88	*2'	049 60	*if flg	077 43	RCL	105 09	9
022 10	*E'	050 02	2	078 01	1	106 44	SUM
023 48	*EXC	051 78	*5'	079 08	8	107 01	1
024 01	1	052 20	*1/x	080 56	*rtn	108 08	8
025 09	9	053 46	*LBL	081 46	*LBL	109 48	*EXC
026 02	2	054 78	*5'	082 15	E	110 01	1
027 41	GTO	055 22	INV	083 50	*st flg	111 08	8

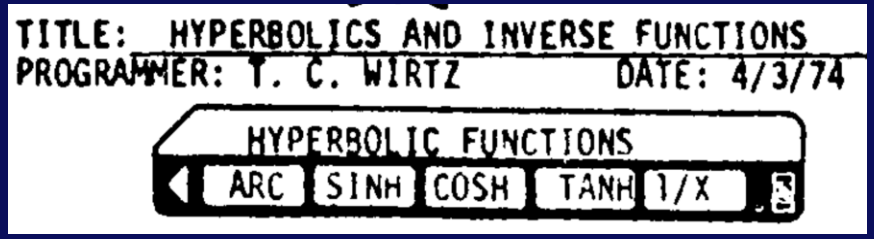
PROGRAM LISTING

DISPLAY	KEY	DISPLAY	KEY	DISPLAY	KEY
112 22	INV	140 01	1	168 01	1
113 49	*PROD	141 09	9	169 22	INV
114 01	1	142 40	*x^2	170 50	*st flg
115 09	9	143 42	STO	171 02	2
116 93	.	144 01	1	172 56	*rtn
117 05	5	145 08	8	173 46	*LBL
118 48	*EXC	146 01	1	174 87	*1'
119 01	1	147 94	+/–	175 22	INV
120 09	9	148 46	*LBL	176 60	*if flg
121 23	Inx	149 69	*9'	177 02	2
122 49	*PROD	150 44	SUM	178 68	*8'
123 01	1	151 01	1	179 20	*1/x
124 09	9	152 08	8	180 46	*LBL
125 48	*EXC	153 48	*EXC	181 68	*8'
126 01	1	154 01	1	182 42	STO
127 09	9	155 08	8	183 01	1
128 41	GTO	156 30	*$\sqrt{x}$	184 09	9
129 50	*st flg	157 44	SUM	185 40	*x^2
130 46	*LBL	158 01	1	186 42	STO
131 88	*2'	159 09	9	187 01	1
132 22	INV	160 48	*EXC	188 08	8
133 60	*if flg	161 01	1	189 01	1
134 02	2	162 09	9	190 41	GTO
135 67	*7'	163 23	Inx	191 69	*9'
136 20	*1/x	164 46	*LBL	192 00	
137 46	*LBL	165 50	*st flg	193 00	
138 67	*7'	166 22	INV	194 00	
139 42	STO	167 50	*st flg	195 00	

HHC 2026 – SR-52 vs. HP-65 - Hyperbolics

HP-65. 65 Notes V1N5P5 – Ted Wirth

97/100 Steps (97%). 2/2 flags, 6/15 labels, 3/10 memories.



SR-52. Or a Gene written program to do the same thing.

110/224 Steps (49%). 0 flags, 9/72 labels, 3/20 memories

GW-01	HYPERBOLIC FUNCTIONS			GW-01
◀ A	HYPERBOLIC FUNCTIONS			GW-01
asinh x	acosh x	atanh x		
sinh x	cosh x	tanh x		1/x

HP-65 HHC 2026 – HP-65 vs. SR-52 – Hyperbolics SR-52

f-1	E	RTN	GTO
SF1	+	LBL	0
f-1	g x<>y	A	LBL
SF2	g LSTx	RTN	D
R/S	-	LBL	C
LBL	f	C	1
E	TF1	f	g x<>y
g	g x<>y	SF1	+
1/x	g NOP	LBL	1
RTN	f-1	B	g LSTx
LBL	TF2	C	-
D	CLx	ENT	/
f	2	f-1	f
SF2	/	SQRT	LN
LBL	GTO	1	2
B	0	f	/
f	LBL	TF1	LBL
SF1	C	CHS	0
LBL	g RUP	g NOP	RCL 8
C	STO 8	+	RCL 7
C	g RUP	f	RCL 6
f-1	STO 7	SQRT	g RUP
LN	g RUP	+	g NOP
ENT	STO 6	f	g NOP
ENT	g RUP	LN	g NOP

000	LBL	035	+	070	INV	105	=
001	A'	036	1	071	LNX	106	RTN
002	STO	037	=	072	STO	107	LBL
003	9	038	DIVIDE	073	9	108	E
004	8	039	(	074	9	109	1/X
005	X^2	040	1	075	+	110	RTN
006	+	041	-	076	LBL		
007	1	042	RCL	077	COS		
008	=	043	9	078	1/X		
009	LBL	044	8	079	=		
010	SIN	045	)	080	DIVIDE		
011	SQRT	046	=	081	2		
012	+	047	LNX	082	=		
013	RCL	048	DIVIDE	083	RTN		
014	9	049	2	084	LBL		
015	8	050	=	085	C		
016	=	051	RTN	086	X		
017	LNX	052	LBL	087	2		
018	RTN	053	A	088	=		
019	LBL	054	STO	089	INV		
020	B'	055	9	090	LNX		
021	STO	056	8	091	STO		
022	9	057	INV	092	1		
023	8	058	LNX	093	9		
024	X^2	059	STO	094	RCL		
025	-	060	9	095	9		
026	1	061	9	096	9		
027	=	062	-	097	+		
028	GTO	063	GTO	098	1		
029	SIN	064	COS	099	=		
030	LBL	065	LBL	100	1/X		
031	C'	066	B	101	X		
032	STO	067	STO	102	RCL		
033	9	068	9	103	1		
034	8	069	8	104	9		

HHC 2026 – In praise of the SR-52

Great things vs the HP-65?

- 224 steps vs. 100 steps.
- Program line numbers!
- BST key.
- 22 data memories vs. 9-ish.
- 72 labels vs. 15.
- Indirect addressing.
- Exchange X and memory.
- Degrees <> radians conversion
- Error detection in a program
- $395 vs. $795

HHC 2026 – In praise of the SR-52

DUMB things vs the HP-65? – besides AOS?

- Register 00 is used for:
  - P/R conversion and DSZ looping.
  - Also likely the best register for IND.
- Comparison tests are on “X”
  - Destructive tests required for X Y comparisons.
- STO 01 requires THREE steps.
- GTO 123 requires FOUR steps.
- Missing Octal and Grads (eh)

HHC 2026 – In praise of the SR-52

Really DUMB things vs the HP-65?

- Missing DMS / HMS arithmetic
- Missing absolute value
- Missing integer / fraction part function!
- Battery contacts with exposed circuit board
- Monstrous size !
- Kahan wrote article on SR-52 arithmetic.
- Happy in some places vs. HP, but not in others.

HHC 2026 – In praise of the SR-52

HP-65 Libraries	SR-52 Libraries
Aviation	Aviation
Chemical Engineering	-----
EE 1 and 2	EE
Finance	Finance
Machine Design	-----
Math 1 and 2	Math
Stat 1 and 2	Statistics
Medical	-----
Stress Analysis	-----
Surveying	Surveying

Games !

HHC 2026 – In praise of the SR-52

HP-67 Libraries

Civil Engineering
EE
Business Decisions
Mechanical Eng
Math
Stat
Clinical Lab / Nuclear
Navigation
Surveying
Games

SR-52 Libraries

Aviation

EE
Finance

Math
Statistics

Surveying
Games

HHC 2026 – SR-52 vs. HP-67 – Gamma Function

HP-67. Viktor Toth - rskey.org/hp97

150 steps. 67%. 2 Memories.

SR-52. Viktor Toth - rskey.org/sr52

221 steps. 99%. 2 Memories.

Of course, from a programming perspective, the HP-67 outperforms the SR-52 on a number of problems.

HHC 2026 – SR-52 vs. HP-67

In most ways, the HP-67/97 > SR-52, but not in all. Like what?

1) The SR-52 could use any register for indirect addressing.

- Memory operations, GTO and SBR could use any register as the indirect object.
- The HP-67/97 could only use the I register.

That's actually a big deal.

2) Data memories - relatively comparable with SR-52 ahead.

- 22 easily accessible registers on the SR-52 vs. 15 easily accessible on the HP-67/97 with 10 swappable through P<>S.

HHC 2026 – In praise of the SR-52

So, what to make of this machine?

As RJN said, it seemed “incomplete” and “rushed”

Despite coming out nearly 18 months after the 65.

Included some good features, but hobbled by poor physical design choices.

IMO, the SR-52 “beat” the HP-65 (Yes, yes AOS).

Importantly, kept the HP / TI competition going.


AND NOW BACK TO
OUR REGULARLY
SCHEDULED RPN
PROGRAMMING