

TI-58C: Changes and Improvements compared to TI-58/59

The TI-58C is a TI-58 with constant memory: its programs and data survive power-off. Its firmware is a revision of the TI-59/TI-58 firmware, not a new design. This document describes what the TI-58C ROM does differently, and why.

Addresses are TI-58C ROM addresses unless written "TI-59 XXXX". They refer to the [annotated TI-58C ROM listing](#) in the Calc-U-59 repository. Statements marked *derived* come from reading the code; nothing observed shows them directly.

Overview

	TI-59 / TI-58	TI-58C
ROM chips	TMC0582, TMC0583, TMC0571B	CD2400, CD2401 (TMC0580), TMC0573
Date code on the dumped chips	1979	1980
User RAM	serial RAM (TMC0598), RAM_OP bus word	CMOS constant memory, MEMWR / MEMRD, 64 raw registers
Registers	120 (TI-58: 60)	60 user registers + 4 extra cells E000–E003
Default partition	479.59 (TI-58: 239.29)	239.29
Card reader	yes (TI-58: code present, no reader)	no, code removed
Printer-present line	KP at digit 0	KP at digit 10
Idle instructions	¼ speed	full speed

How much is shared.

- 4,953 of the 6,144 ROM words align with the TI-59 in order. Another 505 words are TI-59 code that was moved, word for word apart from the jump targets.
- The remaining ~690 words are new or rewritten. Most of them are the former card-reader area 1680–17FF.
- The arithmetic, the transcendental functions, the display formatter, AOS, the program scanner and the print mnemonic table are all unchanged.

The 16 numeric constants and all keycode sub-programs except two are byte-identical. The two that differ are bug fixes; see below.

Why there are more differences than the hardware explains

On paper only three things change: a different RAM chip, a different clock chip, and no card reader. Even so, about a fifth of the ROM differs. These causes compound:

1. Every RAM access site had to be rewritten.

- The TI-59 addresses its serial RAM with one bus word. It carries the operation in nibble 0 and a BCD register number, plus a hexadecimal tens nibble for registers ≥ 100 . Block operations, such as clearing ten registers, are a single word.
 - `MEMWR` / `MEMRD` take a *binary* address (`Sout[1]·16 + Sout[0]`) and move one register at a time.
 - So every place that touches RAM gained a BCD $\rightarrow$ binary conversion: data registers, program registers, LRN store, Ins/Del, label search, CMs/CP, Op 09 and Op 17.
2. **Branch range.** The new code is longer, so routines no longer fit where they were.
 - Whole TI-59 routines were moved to free space: `PGM_SELECT`, Op 16/17, the return-stack push, the tan kernel, HIR, INV List, the library label search, and others.
 - Jumps that now cross too far go through one- or two-word trampolines. The spare words of the print-mnemonic table hold many of them.
 3. **The printer-present line moved to digit 10.**
 - The idle loop's key scans see digit 10 as keyboard row 10, so an attached printer looks like a held key. The idle loop was rebuilt around this (see "Idle loop").
 - The trace gate, List, Adv, Prt and the new Op 40 all test the line at the new digit.
 4. **The card-reader code went away.** That freed most of 1680–17FF, which now holds a factory self-test and relocated routines.
 5. **The ROM is a later revision.**
 - Its dumped chips carry 1980 date codes; the TI-59 set carries 1979.
 - TI used the revision to fix six firmware bugs, two of them in the keycode sub-programs, and to add Op 40 and a TRACE marker.
 6. **Fix and the partition persist.** Constant memory keeps them across power-off. Fix and Op 17 therefore write them into RAM, and power-on restores them.

Constant memory

Addressing

- `MEMWR`: the instruction sends the address on the bus, and the IO value of the *next* instruction is written, within that instruction's field. If the next instruction drives no IO, zero is written. CMs, CP and the first-power-up clear use this to clear registers.
- `MEMRD` puts the register on the bus for the next load (`D=LOAD` , `A=LOAD`).
- **Data register nn:**
 - `REG_ADDR ( 0015 )` computes the RAM index 59 – nn in BCD, as the TI-59 does with its own top register.
 - `RAM_ADDR ( 0096 )` then counts that index down in BCD while it counts `A[1..0]` up in binary.
 - `RAM_ACCESS ( 0A1A )` dispatches to read (`0D9E`), write (`0DFE`) or the LRN path.
 - Ind is decided after a normal read (`0DA3  $\rightarrow$  0F5D`).
- **Program register n** is raw register n:
 - fetch `0B7D–0B85  $\rightarrow$  0EDA`
 - label search `0BDC–0BE3  $\rightarrow$  0F74`
 - LRN write-back `0C16–0C1F`
 - Ins/Del and Op 09 write `1224–122E  $\rightarrow$  0F6A`

- **Cost:** the count loop makes low register numbers slowest. RCL 00 takes 691 active instructions, RCL 29 takes 594; on the TI-59 any RCL takes about 466.

E000: the system register

Raw register 60 (0x3C), called E000 here, holds a signature plus the two settings that survive power-off:

Nibbles	Content	Written at
15–3	signature: the mantissa of constant 0, $\ln 10 = 2302585092994$	first power-up, 037E–0380
2–1	partition $\div 10$ (BCD)	Op 17, 17AE–17B5
0	Fix setting ($n + 2$, 0 = floating)	Fix / INV Fix, 0474–0481

The address is built from SCOM[9] nibble 1 and 0xC. On the TI-58C, SCOM[9] nibble 1 is the high nibble of E000's address (3). On the TI-59 it counts RAM chips. E001–E003 (raw 61–63) are only touched by the first-power-up clear and the self-test.

Power-on (0350–03A6)

Steps from 0000 to the idle loop:

1. **Clear SCOM.** 0000 clears all 16 SCOM rows, as on the TI-59. SCOM loses its contents at power-off, so this runs on every power-on.
2. **Settle delay.** 0352–0355 count a 51-pass loop of about 100 active cycles before the first RAM access. It has no TI-59 counterpart. *Derived* purpose: to let the CMOS RAM interface settle.
3. **RAM-size probe.** 035C–0369 write the signature value to raw register 0x7F and read it back.
 - A 64-register RAM reads back 0. E000 is then 0x3C, and the clear loop makes 64 passes.
 - A 128-register RAM keeps the value. E000 is then 0x7C, the partition becomes 6, the top register 119, and the clear loop makes 128 passes.
 - So the ROM also supports a TI-59-sized CMOS RAM: with a patched emulator it starts up as 479.59, and STO 99 works. This replaces the TI-59's RAM_OP probe of register 90.
4. **Signature check.** 036A–036F compare nibbles 3–15 of E000 with the $\ln 10$ mantissa. Nibbles 0–2 are not validated (*derived*).
5. **First power-up.** If the signature does not match (new batteries, or a corrupted E000):
 - 0370–0379 write zero to all 64 raw registers, E000–E003 included.
 - 037A–0380 write E000 with the signature, partition 3 and Fix floating.
6. **Restore.** 0381–03A6 rebuild the system SCOM rows from E000:
 - SCOM[9]: partition, and the E000 address nibble.
 - SCOM[13]: partition and top register 059.
 - SCOM[0]: Fix.

Then the code falls into CLR.

The TI-58C does not back up the rest of SCOM. There are no MEMWR / MEMRD instructions in the idle or pre-idle path; the only keys that write RAM are Fix, STO-type operations and LRN stores. A power

cycle traced after Fix 2, Rad, St flg 3, Eng, 4 Op 17, a program, an LRN position, Pgm 02 and a pending 7 + 3 (gives:

Survives	Lost
program and data registers	angle unit (back to DEG)
partition	Eng/Sci
Fix	user flags
	pending AOS operations and parentheses
	t, X
	step pointer (LRN shows 000)
	return stack
	selected module program

Idle loop and timing

Printer line. With a printer attached, the KP line at digit 10 is active all the time. The rebuilt idle loop (062A-0659 , plus 0A25-0A30) handles this:

- The release gate masks KP at rows 10, 11, 13 and 14. It runs its window twice, counted by a new toggle fA[11].
- The arming and confirmation scans start at row 12. Rows 10 and 11 go to 0A25 , which rescans rows 9-1 and then re-enters the confirmation scan.
- The TI-59 card-switch poll (TI-59 063C-0640) is gone.

Clock. On the TI-59 an instruction in the idle state takes four times as long as an active one; on the TI-58C idle instructions run at full speed. The ROM compensates in one place only:

Activity	TI-59	TI-58C	Compensated?
Error blink	4 counts per lit pass, both phases $\approx$ 450 ms with printer: period 0.90 s; stand-alone 0.45 s (both measured on a real TI-59)	1 count per pass, 100 + 100 passes $\approx$ 400 + 533 ms: period 0.933 s (0.920 s measured on a real TI-58C with a 389.4 kHz clock, which the model gives at that clock); with printer 333 + 400 ms	yes: the lit-phase +3 was dropped; the period stays $\approx$ 0.9 s
Steady idle pass	9.0 ms (18.0 ms with printer)	4.0 ms (6.7 ms with printer)	—
Key debounce gap (one digit sweep)	4.5 ms	1.33 ms	no
Pause (DISPLAY_PEEK 04E2 , unchanged)	432 ms	128 ms	no
GTO peek while running	216 ms	64 ms	no

The TI-59 blink and Pause figures were checked against a real TI-59 filmed next to a stopwatch. On the TI-59 the printer-present line sits at D0, which its idle scan sees as a "row 0" key and discards: two extra digit sweeps per pass with a printer. The TI-58C times are computed from the emulator's model: $384 \text{ kHz} \div 32 = 12,000$ instructions per second (the TI-58's part, assumed), idle at full speed. Real units vary: one measured PHI 194.7 kHz (389.4 kHz) and blinks 0.920 s, another blinks 0.896 s (as at 400 kHz); each unit's other measurements are compared at the clock its blink gives. The short Pause is real: a filmed TI-58C (the 0.896 s unit) shows each Pause lit for about 0.122 s; the model gives 0.118–0.120 s at that unit's clock.

Bug fixes

Each fix can be reproduced in Calc-U-59 by switching the model between TI-59 and TI-58C: the TI-59 shows the bug, the TI-58C does not.

#	TI-58C	TI-59	Bug on the TI-59	Example
1	03E3 CLR fA[7]	03DB	An Ind pending from run mode survives into LRN. The first recorded key gets an extra 00 step.	Ind LRN 5 stores 00 05 (TI-58C: 05); Ind LRN ST0 12 stores 00 42 12
2	0920 CLR fA[7]	092E	Ind List starts one step late and garbles operands.	program STO 12 lists "001 12 B" instead of "000 42 STO / 001 12 12"
3	0A60–0A62	0A5B–0A5E	A keyboard $\Sigma+$, $\bar{x}$, D.MS, P→R or Op 11–15 pushes the step counter unchanged. After a halt or SST (program-register cache valid, fB[11] clear) that counter names the step last executed, so the next R/S runs that step again.	program 1 2 3 4 5 R/S, keys RST SST SST $\Sigma+$ R/S show 2345 (TI-58C: 345)
4	0D30 CLR fA[10]	—	With an Ind pointer register outside the partition, the pending trace record (fA[10]) suppresses the "?" error line, so the error goes untraced. Affects STO/RCL/SUM/Prd/Exc/Fix/Op/GTO/SBR/Pgm Ind.	printer TRACE on: 5 ST0 Ind 65 prints no "5. ?" line
5	12E3 MOV R5,fB[1..4]	11BD (fA)	HIR SUM/Prd/INV SUM/INV Prd take the operand's exponent sign from the display flags (fA[3]). These are stale in a running program and clear in a non-sci display, so $0 < x < 1$ is used as $m \cdot 10^+ + e $. HIR only reaches a RAM program	3 HIR 08 .5 HIR 48 HIR 18 gives 150 (TI-58C: 1.5)

#	TI-58C	TI-59	Bug on the TI-59	Example
			through pseudo-codes, the Ind merge or cards. The keycode sub-programs are not affected.	
6	keycode step 031	step 031	Op 12/14/15: a CE right after the slope clears the error of a 0/0 or x/0 slope (all x equal, or one data point). The TI-58C has CP there instead, which only clears t (overwritten by the next $x \rightleftharpoons t$), so the error reaches the result.	$2 \ x \rightleftharpoons t \ 1 \ \Sigma + \ 2 \ x \rightleftharpoons t \ 3 \ \Sigma +$ $Op \ 12$ shows a steady $0 / t = 1$ (TI-58C: flashing)
7	keycode steps 251–281	251–283	$R \rightarrow P$ (INV $P \rightarrow R$) ends with a CE meant for the deliberate 0/0 of the quadrant trick at $x = 0$. It also hides an overflow of $x^2 + y^2$. The TI-58C computes θ first, clears, then computes r , and no longer clobbers Hir 7.	$3 \ x \rightleftharpoons t \ 2 \ EE \ 50 \ INV \ P \rightarrow R$ gives $r = 1E50$ without an error (TI-58C: flashing)

Fix 7 has a side effect: an underflowing $x^2 + y^2$ (for example $y = 2E-60$) now flashes, although r is correct. The new $R \rightarrow P$ also has an extra CE at step 252 with no observable effect.

Enhancements

- **Op 40** (08D8 $\rightarrow$ 0BF8) sets flag 7 if a printer is connected. Otherwise flag 7 is left unchanged, and INV has no effect. On the TI-59 every Op 40–99 is an error; on the TI-58C, Op 41–99 still are.
- **TRACE marks taken branches** (15F0 $\rightarrow$ 17D4-17E9 $\rightarrow$ 14EB). When $x=t$, $x \geq t$ or If flg transfers, the address line prints "▲" (PC-100 code 61) before the address, so $1 \ x \geq t \ 006$ prints "▲6".
 - Branches not taken are printed without the marker.
 - Dsz, Ind operands, GTO and SBR never get it.
 - The test reads the continuation slot in SR (0x0C0, 0x0BC, 0x0B8) and fB[8].
- **Factory self-test.**
 - **Entry:** the idle loop polls test lines (064B: KN at digit 15; KN on row 8 or 9), which a test fixture drives.
 - **Sequencer** (0DB7–0DDA): prints the test numbers 1–11 and dispatches through the table 0C01–0C0A. Test numbers 3 and 7 have no body.

Test	Body	Checks
1	09CC	constant-ROM entry 7
2	08BB	constant-ROM entry 63
4, 5	16C7, 16C8	SCOM rows 0–7 and 8–15, with a 6969.../9696... checkerboard

Test	Body	Checks
6	1680	all 64 raw RAM registers, with the same checkerboard; prints "D" first on a 128-register RAM
8, 9	1360 , 1368	SCOM rows written with their own index
10	1375	RAM[k] := k for all 64 registers

- **Result:** pass goes 10BF → 0DC3 ; fail goes 10BE → 0DD6 → 09DD , which ends in a cold restart.
- **Side effect:** the test wipes SCOM and all of constant memory, E000 included.
- **Open:** a KR line at digit 10 (064E → 09F7) jumps to 0x1800, beyond the 6K ROM. Possibly a hook for test-fixture code; unexplained.
- **Fix and the partition survive power-off** (see "Constant memory").

Kept from the TI-59 (still there)

- **Fast mode.** The St flg / KEY_ENCODE mis-dispatch at 0B34 and the HIR STO into SCOM[0] (11BC) work exactly as on the TI-59 ([TI-59 Fast Mode](#)). The 69! test program runs in 363k weighted cycles in fast mode against 585k normally.
- **The 7th SBR level is still dropped.** The push moved to 1334 , and the depth check is at 1339-133C .
- **Unchanged behaviour:**
 - the Pgm, library label search and library transfer handling;
 - Op 09, and Op 09's refusal of protected modules;
 - LIB_RETURN, label search, Dsz, CMs/CP semantics, List, INV List and Adv.
- CRD.OFF (0005) is still issued at reset, with no effect.

Removed

- **The card reader:**
 - TI-59 1680-17AB (CARD_IO);
 - the card gate and Write entry (TI-59 0459-0461);
 - the idle-loop card-switch poll.

Write (slot 0069) now only redisplay X (5 Write shows 5.).
- **Memory protection,** which on the TI-59 a protected card sets in SCOM[9] nibble 2. Its tests are gone from LRN, List, the trace gate, the GTO peek, Op 17 and CP
- **The RAM_OP bus word.** Its block clears are replaced by loops.