CS617 Chapter 09

Chapter 09 | Native MathROM: 23 Function Contracts

Edition r003 | 2026-11-08 | Casio PB-2000C | Assembly reference
Scope: 23 native THIN wrappers that invoke the resident mathematical services in the Casio PB-2000C ROM.

How to use this book

Every entry identifies the Casio ROM service, its wrapper, the inputs and outputs used by the Chapter 09 program, and a matching expression to try in MREVAL. The MREVAL expression is an interactive demonstration; assembly programs call the corresponding `MATH_*` entry.

Shared call pattern: Include `math_name.inc` once in the assembly source; initialize the appropriate registers and PB-2000C machine state; execute `CAL MATH_NAME`; read the result in its documented representation. The wrapper uses `JP &Hxxxx` to enter the resident Casio ROM.

Quick index

Label	Operation	Resident ROM address	Result
`MATH_ACS`	Inverse cosine	`&H0993`	B9
`MATH_ADD`	Floating addition	`&H068C`	B9
`MATH_ASN`	Inverse sine	`&H0991`	B9
`MATH_ATN`	Inverse tangent	`&H0995`	B9
`MATH_B2BCD`	Binary integer to packed BCD	`&H0B53`	packed BCD
`MATH_CMP`	Compare two B9 values	`&H0B13`	C/Z flags
`MATH_CNVR`	Signed 16-bit integer to B9	`&H0B38`	B9
`MATH_COS`	Cosine	`&H0926`	B9
`MATH_DIV`	Floating division	`&H06F8`	B9
`MATH_EXP`	Natural exponential	`&H08EE`	B9
`MATH_FIX`	Integer-part operation	`&H0A82`	B9

Label	Operation	Resident ROM address	Result
MATH_INC	Increment by one	&H068A	B9
MATH_LGT	Base-10 logarithm	&H08B9	B9
MATH_LOG	Natural logarithm	&H08B7	B9
MATH_MUL	Floating multiplication	&H06B9	B9
MATH_PI	Load PI constant	&H0A77	B9
MATH_POWER	Power helper	&H09E0	B9
MATH_RND12	Rounding service	&H0A7B	B9
MATH_SIN	Sine	&H0924	B9
MATH_SQ	Square	&H06B5	B9
MATH_SQR	Square root	&H071F	B9
MATH_SUB	Floating subtraction	&H0686	B9
MATH_TAN	Tangent	&H092A	B9

Detailed member contracts

MATH_ACS - Inverse cosine

Native target: `&H0993` | **Include:** `math_acs.inc`
Call: `CAL MATH_ACS` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `acs(0.5)`. An inverse-trigonometric service; the returned unit depends on machine angle state.
- **Implementation considerations:** Angle mode; +/-1 endpoints; out-of-domain inputs; register and scratch clobbers.

MATH_ADD - Floating addition

Native target: `&H068C` | **Include:** `math_add.inc`
Call: `CAL MATH_ADD` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`; `y` B9-expanded-9-byte at `GPR $00..$08`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `add(12,5)`. Adds first and second expanded-B9 operands; expected illustrative result 17.
- **Implementation considerations:** Operand order/aliasing, cancellation, exponent carry, overflow, scratch clobbers.

MATH_ASN - Inverse sine

Native target: `&H0991` | **Include:** `math_asn.inc`
Call: `CAL MATH_ASN` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `asn(0.5)`. Inverse-trigonometric service; inspect angle-state conventions before interpreting the result.
- **Implementation considerations:** Angle mode, endpoints, out-of-domain values and clobbers.

MATH_ATN - Inverse tangent

Native target: `&H0995` | **Include:** `math_atn.inc`
Call: `CAL MATH_ATN` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `atn(1)`. Inverse-trigonometric service with angle-state dependency.
- **Implementation considerations:** Sign, extremes, angle mode and clobbers.

MATH_B2BCD - Binary integer to packed BCD

Native target: `&H0B53` | **Include:** `math_b2bcd.inc`
Call: `CAL MATH_B2BCD` (after caller setup).

- **Input:** `x` integer-16-bit-signedness-open at `GPR $03..$04`.
- **Output:** `result` packed-BCD-3-byte at `GPR $14..$16`.
- **Explorer lesson:** `b2bcd(123)`. Produces packed BCD, not expanded nine-byte B9. Never decode its output using B9 conventions.
- **Implementation considerations:** Signedness and byte order of $03/$04; packed $14-$16 layout; zero, extrema, aliasing.

Format example: The three packed-BCD bytes in `$14..$16` are not an expanded B9 result; use the packed format.

MATH_CMP - Compare two B9 values

Native target: `&H0B13` | **Include:** `math_cmp.inc`
Call: `CAL MATH_CMP` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`; `y` B9-expanded-9-byte at `GPR $00..$08`.
- **Output:** `result` flags-CZ at `CPU flags C/Z`.
- **Explorer lesson:** `cmp(1,2)`. Produces a CPU flag result, not a nine-byte floating-point number.
- **Implementation considerations:** C/Z capture immediately after return, signed ordering, flag clobbers, stack and RAM scratch.

Comparison example: the Chapter 09 emulator reports classify 2<3 as carry, 3>2 as neither carry nor zero, and 2=2 as zero. Read C/Z before executing an instruction that changes flags.

MATH_CNVR - Signed 16-bit integer to B9

Native target: `&H0B38` | **Include:** `math_cnvr.inc`
Call: `CAL MATH_CNVR` (after caller setup).

- **Input:** `x` signed-16-bit-word at `GPR $15..$16`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `cnvr(123)`. Takes signed integer in $15/$16 (byte order needs contract qualification) and returns B9.
- **Implementation considerations:** Zero, negative, int16 extrema, sign extension, exact B9 bytes and clobbers.

MATH_COS - Cosine

Native target: `&H0926` | **Include:** `math_cos.inc`
Call: `CAL MATH_COS` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `cos(60)`. Trigonometric call; MREVAL forces DEG for illustrative expression. Direct callers must establish angle mode.
- **Implementation considerations:** Quadrants, unit/mode, rounding, magnitude and clobbers.

MATH_DIV - Floating division

Native target: `&H06F8` | **Include:** `math_div.inc`
Call: `CAL MATH_DIV` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`; `y` B9-expanded-9-byte at `GPR $00..$08`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `div(8,2)`. The resident division service is at 06F8h, distinct from the 06F6h helper.
- **Implementation considerations:** Zero divisor in raw native call, tiny/large quotients, operand order, under/overflow, clobbers.

Division example: MREVAL checks the divisor and displays an error for `div(8,0)` before invoking the resident DIV service.

MATH_EXP - Natural exponential

Native target: `&H08EE` | **Include:** `math_exp.inc`
Call: `CAL MATH_EXP` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.

- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `exp(1)`. Computes the exponential using the resident implementation.
- **Implementation considerations:** Exponent extremes, overflow/underflow, B9 rounding and clobbers.

MATH_FIX - Integer-part operation

Native target: `&H0A82` | **Include:** `math_fix.inc`
Call: `CAL MATH_FIX` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `fix(7.5)`. Removes or treats the fractional portion under resident rules; do not assume all negative cases.
- **Implementation considerations:** Negative fractions, rounding/truncation details, integer extremes and clobbers.

MATH_INC - Increment by one

Native target: `&H068A` | **Include:** `math_inc.inc`
Call: `CAL MATH_INC` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `inc(7)`. Native increment service, not a syntactic macro for ADD.
- **Implementation considerations:** Signed values, carry, exponent extremes, input aliasing and clobbers.

MATH_LGT - Base-10 logarithm

Native target: `&H08B9` | **Include:** `math_lgt.inc`
Call: `CAL MATH_LGT` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `lgt(100)`. Common logarithm (base ten), distinct from natural LOG.
- **Implementation considerations:** Zero/negative domain, unity, boundary values, accuracy and clobbers.

MATH_LOG - Natural logarithm

Native target: `&H08B7` | **Include:** `math_log.inc`
Call: `CAL MATH_LOG` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.

- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `log(2)`. Natural logarithm (base e), distinct from base-ten LGT.
- **Implementation considerations:** Zero/negative domain, unity, extremes and clobbers.

MATH_MUL - Floating multiplication

Native target: `&H06B9` | **Include:** `math_mul.inc`
Call: `CAL MATH_MUL` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`; `y` B9-expanded-9-byte at `GPR $00..$08`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `mul(6,7)`. Multiplies two expanded B9 values.
- **Implementation considerations:** Zero/sign handling, under/overflow, operand overlap and scratch clobbers.

MATH_PI - Load PI constant

Native target: `&H0A77` | **Include:** `math_pi.inc`
Call: `CAL MATH_PI` (after caller setup).

- **Input: No explicit operand**.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `pi()`. No explicit operand is required; output is resident B9 constant, not proof that nothing else changes.
- **Implementation considerations:** Exact nine-byte oracle, mode sensitivity and all register/flag side effects.

No operand: `MATH_PI` takes no numeric argument. As with other resident calls, preserve any caller state your application needs.

MATH_POWER - Power helper

Native target: `&H09E0` | **Include:** `math_power.inc`
Call: `CAL MATH_POWER` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`; `y` B9-expanded-9-byte at `GPR $00..$08`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `power(2,3)`. A low-ROM power helper; distinguish it from language-card power adapters.
- **Implementation considerations:** Dependency path, negative bases, fractional powers, 0^0, overflow and clobbers.

MATH_RND12 - Rounding service

Native target: `&H0A7B` | **Include:** `math_rnd12.inc`
Call: `CAL MATH_RND12` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `rnd12(12.345)`. Resident twelve-digit rounding service; not a random-number generator.
- **Implementation considerations:** Tie handling, carry into exponent, negative inputs, and true rounding policy.

MATH_SIN - Sine

Native target: `&H0924` | **Include:** `math_sin.inc`
Call: `CAL MATH_SIN` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `sin(30)`. Trigonometric service; direct use requires explicit angle mode.
- **Implementation considerations:** Quadrants, large angles, zero, angle mode, rounding and clobbers.

MATH_SQ - Square

Native target: `&H06B5` | **Include:** `math_sq.inc`
Call: `CAL MATH_SQ` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `sq(7)`. Square a value; SQ is NOT square root.
- **Implementation considerations:** Negative/zero inputs, overflow, numeric precision and clobbers.

MATH_SQR - Square root

Native target: `&H071F` | **Include:** `math_sqr.inc`
Call: `CAL MATH_SQR` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `sqr(9)`. Square root; SQR is NOT square.
- **Implementation considerations:** Negative, zero and extreme inputs; mode/error conventions and clobbers.

MATH_SUB - Floating subtraction

Native target: `&H0686` | **Include:** `math_sub.inc`
Call: `CAL MATH_SUB` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`; `y` B9-expanded-9-byte at `GPR $00..$08`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `sub(12,5)`. Subtract second B9 operand from first; do not assume operand order without its register map.
- **Implementation considerations:** Cancellation, negative results, aliasing, operand order and clobbers.

MATH_TAN - Tangent

Native target: `&H092A` | **Include:** `math_tan.inc`
Call: `CAL MATH_TAN` (after caller setup).

- **Input:** `x` B9-expanded-9-byte at `GPR $10..$18`.
- **Output:** `result` B9-expanded-9-byte at `GPR $10..$18`.
- **Explorer lesson:** `tan(45)`. Trigonometric service; singularities and angle state matter.
- **Implementation considerations:** Near-pole behavior, periodic reductions, angle mode and clobbers.

One complete calling example: CNVR

The resident CNVR service converts a 16-bit signed integer into the expanded nine-byte B9 representation. This section shows the difference between **the wrapper** and **the caller**. The example follows the `MATH_CNVR` path exercised in the Chapter 09 MathABI probe.

The actual source of `math_cnvr.inc` (executable portion):

```
MATH_CNVR:
        JP      &H0B38
```

In the consuming application:

```
; Include the wrapper once among the program's includes.
#INCLUDE (lib/math/math_cnvr.inc)

; Later, from code running in the initialized PB-2000C environment:
        LDW     $15,1        ; input: signed 16-bit value +1
        CAL     MATH_CNVR    ; enters Casio's resident service
        ; On return: expanded B9 result in $10 through $18.
        ; Copy/use the result before another call overwrites it.
```

`CAL` creates the application's return frame; the wrapper's `JP` transfers directly to the resident code, which returns to the original caller. `LDW $15,1` and `CAL MATH_CNVR` are taken from the `MathABI R003a` calling fixture. These lines are **an excerpt**, not a standalone cartridge: a complete program must provide the platform header, startup, stack and firmware environment as in the Chapter 09 programs. The bytes in `$10..$18` are B9, not ASCII text; use the normal B9 formatting support to display them.

Three additional MREVAL commands

Explorer command	Implementation origin	Library role
`comp(a,b)`	Derived signed compare adapter from `MATH_CMP`	Not an extra original low-ROM service
`sign(x)`	Shared `B9_SGN`	Not an original low-ROM service
`cordic(i)`	Historical seven-byte table read, `i` 0..21	Not a CORDIC iterative algorithm

Further reading

- VT - Virtual Terminal User's Guide
- WHAT r007 - Native MathROM THIN Library and MREVAL Explorer
- Native MathROM THIN ABI and Evidence Register R001
- MathROM THIN / Math 1.1 Source Identity and ABI Proof-Boundary Audit
- Chapter 09 MathABI R003a Five-Report Validation Witness
- Chapter 09 r007 Engineering Evidence and Gap Reconciliation