MC68332 Instruction Set and Practical Reference Guide for Trionic 5 decompialtion

for www.trionictuning.com

By Biohazard_86

0 – README

- This is code in assembly
- This is code in C
- References to tables, pages or other stuff refers to the datasheet of the MC68332.
-

1 - Programming Model and Registers

The MC68332 core provides eight 32-bit data registers (D0–D7) and eight 32-bit address registers (A0–A7). A7 serves as the stack pointer (SP), which has two versions: USP (user) and SSP (supervisor). The Program Counter (PC) is 32-bit. The Status Register (SR) is 16-bit and contains the Condition Code Register (CCR) with bits X, N, Z, V, and C, plus system bits T1:T0 (trace), S (supervisor mode), and IP[2:0] (interrupt mask). VBR defines the base address for the vector table, and SFC/DFC define bus access spaces (user/supervisor).

2 - Addressing Modes — summary and examples

The CPU32 supports classic M68000 addressing modes with some extensions:
• Register direct (Dn): `MOVE.L D0, D1`
• Address register direct (An): `MOVE.L A0, A1`
• Indirect (An): `MOVE.W (A0), D0` (load the word from the address A0)
• Post-increment (An)+: `MOVE.B (A0)+, D0` (reads the byte and A0+=1)
• Pre-decrement -(An): `MOVE.L -(A7), D0`
• Displacement (d16,An): `MOVE.L 8(A3), D0` (reads from A3+8)
• Indexed (d8,An,Xn): `MOVE.L 4(A2,D3.W), D1`
• PC-relative (d16,PC): `LEA my_table(PC), A0`
• Absolute 16/32-bit: `MOVE.L $40000, D0`
• Immediate: `MOVEQ #10, D0`
These modes provide flexible and efficient addressing for tables, stacks, and peripherals.

3 -Flags — SR / CCR

The Status Register (SR) has 16 bits.
The upper bits contain `T1:T0`, S, and `IP2..IP0`. The lower byte, CCR, contains X (extend), N (negative), Z (zero), V (overflow), and C (carry).
Examples:
BEQ — Z=1 (Equal)

BNE — Z=0 (Not equal)

BGT — (Z=0) && ($N \oplus V = 0$) (Greater signed)

BLT — $N \oplus V = 1$ (Less signed)

BCS/BLO—Carry set

BCC/BHS—Carry clear

4 - Instruction groups

A) **Data transfer:**
MOVE — move data (byte/word/long): `MOVE.W (A0), D1` -> D1.low = memory(A0)
MOVEA — move to address register: `MOVEA.L A0, A1`

MOVEM — move multiple: MOVEM.L D0-D3, -(A7)
MOVEQ — quick load: MOVEQ #-1, D0 (load 0xFFFFFFFF)
PEA — push effective address: PEA 8(A3)

B) **Arithmetic:**
ADD / ADDI / ADDQ / ADDA / ADDX:
ASM: ADD.L D2, D1 ; ADDI.W #4, D0 ; ADDA.L D3, A1

SUB / SUBI / SUBQ / SUBA
ASM: SUB.W #2, D0

MULS / MULU (16×16→32 or 32×32)
ASM: MULS.W (A0), D1 results in 32-bit in D1.

DIVS / DIVU
ASM: DIVS.W (A0), D1 ; D1 = D1 / (word in memory)

Example:
```
MOVE.W (A0), D0
MULS.W (A1), D1
```
C equivalent:
```
int16_t a = *(int16_t*)addr0;
 int16_t b = *(int16_t*)addr1;
int32_t r = (int32_t)a * (int32_t)b;
```

C) **Logic and bits:**
AND, OR, EOR, NOT: Example: ANDI.B #0x7F, D0
BTST, BSET, BCLR, BCHG: test / set / clear / toggle bit
ASM: BTST #3, (A0)
Tester bit 3 in memory (A0)
Typical use: access to flags in peripheral registers.

D) **Shifts and rotates:**
ASL/ASR/LSL/LSR/ROL/ROR:
LSR.L #1, D0 divides by 2.

E) **Compare/test:**
CMP, CMPI, CMPA, CMPM:

CHK, CHK2. Example: CMP.W #0, D0.

F) **Flow control:**
BRA, BSR, Bcc, DBcc,
ASM: BSR subroutine ; BEQ label ; DBNE D0, loop

JMP, JSR, RTS, RTE, RTD

TRAP, TRAPV
STOP.

G) **System/Supervisor:**
MOVE to/from SR — MOVE SR, D0 or MOVE D0, SR (the latter is privileged on CPU32).

MOVEC — Move to/from control registers (e.g., VBR, SFC/DFC). Example: MOVEC.L D0, VBR

LINK / UNLK — stack frame management (used by compilers for Prologs/Epilogs).

RESET, BGND, BKPT — testing/debugging support.

5 - CPU32-specific instructions (TBLS/TBLU/TBLSN/TBLUN)

These instructions are very relevant to Trionic: table lookup and interpolation (VE/ignition maps). Operation (summary):

-Form: TBLS <ea>, Dn or TBLU <ea>, Dn

-They operate on two operands representing indices and axes: they use a table in memory and the contents of Dn (or its low byte) to calculate interpolation between two entries (Dym and Dyn in the table) and place the result in Dn.

Conceptual interpretation (steps):

1 - Calculate Temp = Dyn - Dym.

2 - Temp = (Temp * Dn[7:0]) (or similar operation).

3 - Combine to produce the interpolated value and write it to Dn.

Why it's important: In Trionic, you'll see routines with many TBLS/TBLUs—these are typically where injection/ignition maps are evaluated. Identifying them is indicated by the engine control kernel.

6 - Privileged instructions — system indicators

Privileged instructions (supervisor mode only): MOVE to SR, STOP, RTE, MOVEC to VBR/SFC/DFC. These indicate initialization, boot, or ISR routines.
Example: `MOVE.L D0, SR` ; sets interrupt mask.

Helpful tip: Whenever you see MOVE to SR or MOVEC VBR, mark that function as system init; it often prepares peripherals (SIM/QSM/TPU) before jumping to the main loop.

7 - Peripheral access examples (QSM, TPU, SIM)

8 - BDM — commands and examples

BDM (Background Debug Mode) allows you to read registers and memory, perform dumps, and execute small patches in RAM. Command table and descriptions are available in the datasheet (Table 21).

RDREG / RAREG — Read Dn/An register.
WDREG — Write register.
RSREG / WSREG — Read/write system registers (e.g., SR, VBR).
READ addr — Read memory (size defined by the command).
DUMP — Dump large blocks (useful for creating backup_T5.bin).
WRITE / FILL — Write or fill blocks.
CALL — Execute small code loaded into RAM (useful for checksums or boot).
GO — Resume execution.