

P8000EMU vs. MAME 0.123

Difference analysis of a 2008/2009 MAME derivative and comparison with current MAME

Report date: 13 October 2026 | Prepared by: Claude (AI assistant)

Item	Detail
Subject	P8000 computer emulation (Robotron/EAW P8000, 8-bit Z80 board and 16-bit Z8001 board), implemented in 2008/2009 by Matt Knoth as a derivative of MAME v0.123s
Inputs	(1) mame-mame0123.zip: complete original MAME 0.123, 3,325 files under src/. (2) p8000_mame_src.zip: modified sources, 359 files (only those needed to build the P8000 target). (3) Current MAME: mamedev/mame master, commit f26abf2b (13 Oct 2026), sparse checkout of the relevant directories.
Deliverables	This report, the Markdown version of the findings, and a bundle with all 35 per-file diffs plus the lists of changed and new files.
Status of findings	Stage 1 (difference analysis) is complete. Stage 2 (comparison with current MAME) is a first survey: it identifies what exists and what probably covers each fix, but equivalence of individual fixes is not yet proven by testing.

1. Executive summary

- **Size of the change.** Of the 359 files in the P8000 package, 299 are identical to MAME 0.123 (line endings ignored), 35 are modified and 25 are new. The new files comprise 19 P8000-specific source files (about 17,200 lines), a README and five DirectX headers used only for the Windows build.
- **Nature of the change.** The P8000 sources add a segmented Z8001 CPU with a Z8010 MMU, a simplified i8272 floppy controller, a CopyQM image reader, four machine drivers and a terminal/keyboard layer. In the existing code, the author fixed or extended the Z80 DMA, CTC, PIO and SIO peripherals, patched the memory system for byte access in 16-bit I/O space, and reworked the user interface and keyboard input so that MAME behaves like a computer rather than an arcade machine.
- **Header comments are incomplete.** As suspected, the 'modified for P8000EMU' notes do not cover every change. All 35 files were therefore compared line by line; several important fixes (for example in the DMA, CTC and PIO code) carry no comment at all.
- **Current MAME already contains most building blocks.** It has a segmented Z8001 (z8001_device), a Z8010 MMU device, Z80 DMA/CTC/PIO/SIO devices, the i8272a controller, a CopyQM format reader and, importantly, a skeleton driver src/mame/skeleton/p8k.cpp (BSD-3-Clause, 2009) that was explicitly derived from Matt Knoth's emulator. It is flagged *not working*.
- **What is missing.** The Z8010 is not used by the 16-bit driver, there is no equivalent of the 8-bit-board abstraction layer used by the 16-bit board, no Winchester (WDC) controller, and the terminal, keyboard and floppy handling must be re-created with modern MAME mechanisms.
- **Open verification points.** DMA byte counting, the CTC down-counter read-back and instruction-level accuracy of the Z8001 core could differ from the original fixes and need targeted tests.
- **Licensing.** The P8000 driver code is under a non-commercial freeware licence that the author may change retroactively; it is not compatible with MAME's BSD-3/GPL model. Behaviour and hardware knowledge can be re-implemented, the code itself should not be copied.

2. Task and scope

The request came in three stages:

Stage	Question	Status
1	Work out, in detail, the differences the original developer introduced relative to the MAME version the port was based on.	Done (sections 4-6)
2	Determine whether the developer's fixes are already present in current MAME.	First survey done (section 7); detailed verification pending
3	Implement the P8000 system in current MAME and assess whether that is feasible.	Feasibility assessed (section 9); implementation not started

Constraints given by the requester: the modified tree contains only the files needed to build the P8000 target; the original tree is complete; C sources in the modified tree usually carry change comments in their headers, but these must not be relied upon.

3. What was done

Step	Action	Result
1	Extracted both archives into separate, empty directories and counted files.	Original: 3,332 files (3,325 under src/). P8000: 359 files under src/.
2	Identified the base version from the P8000 README and the file timestamps.	Base: MAME v0.123s (original tree is mame0123). Modification dates range from May 2008 to March 2009.
3	Listed all files in both trees; computed the intersection and the files that exist only in the P8000 tree.	334 common files, 25 new files.
4	Compared every common file with <code>diff --strip-trailing-cr</code> (the P8000 tree uses CRLF line endings).	299 identical, 35 changed. Unified diffs were generated for all 35.
5	Read every diff and the headers and entry points of all new files; traced which core hooks the new drivers actually use (e.g. <code>z80ctc_create_interrupt</code> , <code>osd_read_ntv_keyboard</code> , <code>ui_show_*_menu</code>).	Categorisation in sections 5-6.
6	Compared the new Z8001 sources with the original Z8000 sources from MAME 0.123 to estimate how much is derived and how much is new.	<code>z8001ops.c</code> is about 90% identical to <code>z8000ops.c</code> ; segmented operations and table are new.
7	Searched current MAME for each affected component, cloned the relevant directories and inspected the implementations.	Section 7.
8	Wrote the report and bundled the diffs.	This document plus the diff bundle.

Limitations of the method. The comparison is static: no code was compiled or run, because the work environment has no full MAME build and no P8000 ROMs or disk images. Statements such as 'probably included' in section 7 are based on reading the current sources and must be confirmed by tests.

4. Inventory

Category	Files	Remarks
P8000 package, total	359	Sources only; no ROMs, no disk images
Identical to MAME 0.123	299	Line endings ignored (the P8000 tree is CRLF, the original LF)
Modified	35	Listed in appendix A, with lines added and removed
New	25	19 P8000-specific sources, README, 5 DirectX headers (appendix B)

Line counts of the new P8000-specific sources:

Component	Files	Lines
Z8001 CPU (core, operations, segmented operations, tables, disassembler)	7	about 13,060
Z8010 MMU	2	434

Component	Files	Lines
i8272 floppy controller with attached drives	2	625
CopyQM image reader	2	765
Drivers (8-bit board, 16-bit board, abstraction layer, WDC)	4	1,839
8-bit board video/terminal and its header	2	457
Total	19	about 17,180

5. New components in detail

5.1 Z8001 CPU

The CPU is derived from Juergen Buchmueller's Z8002 core in MAME 0.123 (dual MAME/GPL licence) and extended to the segmented Z8001. `z8001ops.c` (6,562 lines) differs from the original `z8000ops.c` (6,503 lines) in about 710 lines. `z8001ops_seg.c` (3,697 lines) is new and contains the segmented-address variants. The opcode table doubles from 596 to 1,212 lines, and the disassembler grows from 318 to 379 lines. A compile-time option `Z8001_CPU_INST_COVERAGE` exists in `cpu.mak`. The core calls `debug_request_update()` when the MMU mapping changes so that the disassembly view stays correct. The original Z8000 core is replaced (CPU_Z8000 becomes CPU_Z8001), not supplemented.

5.2 Z8010 MMU

A compact implementation (259 + 175 lines, GPLv3) supporting up to four instances. The 16-bit driver defines three access classes (code, data, stack) and drives the MMUs for address translation.

5.3 Floppy support: i8272_floppies and cqm_image

`i8272_floppies` models an i8272 controller with up to four attached drives and is deliberately simplified 'to speed up floppy emulation compared to MESS nec765 detailed modelling'. Unimplemented commands are listed as a TODO. `cqm_image` is a port of the CopyQM reader from LIBDSK 1.2.1 (LGPL) and is read-only; it composes image paths from the new `floppy_directory` option.

5.4 Machine drivers

File	Role
<code>p8000_8.c</code> , <code>includes/p8000_8.h</code>	8-bit board (Z80): ROM / static RAM / dynamic RAM banking, CTC, PIO, SIO, DMA, floppy controller, beeper. Hooks the in-game handler for keyboard shortcuts and exit dialog.
<code>video/p8000_8.c</code>	Terminal emulation (24 x 80 character memory, cursor handling, escape sequences), reading of the native keyboard buffer, F-key menus (About, TTY, floppy 0/1, exit).
<code>p8000_16.c</code>	16-bit board (Z8001 + Z8010): three MMU access classes, 1 MB RAM boundary, CTC/PIO/SIO with a Z80 daisy chain, periodic timer that generates CTC interrupts directly.
<code>p8000_8_abstr.c</code>	'Abstraction layer': emulates the 8-bit board for the 16-bit board (terminal, keyboard, floppy access) so that the 16-bit system can boot and use peripherals without a second CPU.
<code>p8000_16_wdc.c</code>	Winchester disk controller for the 16-bit board: command/status handling, a 53 MB RAM-backed disk, parameter table 'WDC_4.2'.

6. Modified files in detail

6.1 Fixes and extensions to existing peripheral devices

File	Change	Purpose
z80dma.c/.h	Interrupt support (daisy-chain functions <i>irq_state/ack/reti</i> , <i>irqcb</i> callback, interrupt on end of block); read sequence (<i>rd_seq</i>) and status byte; WR6 commands 0xA3, 0xA7, 0xAB, 0xAF, 0xBF and 0xC3 implemented; byte counting corrected (counter starts at 0 and is compared with the block length, Load sets count to 0); WR3 bug fixed (the mask byte was scheduled without testing bit 3); reset per channel; port accessors take <i>offs_t</i> .	In 0.123 the DMA existed only for two arcade games; reading a register ended in <i>fatalerror</i> .
z80ctc.c/.h	Counter-mode period without error log; reading a channel in reset state returns the down counter; down-counter read-back recomputed as constant minus elapsed time divided by period; new <i>z80ctc_create_interrupt(which, ch)</i> .	The driver generates the low-frequency interrupts itself instead of simulating input clocks nanosecond by nanosecond (speed).
z80pio.c/.h	Ready callback receives (<i>channel<<1</i>) <i>state</i> ; data type unsigned; interrupt mask logic (<i>data ^= ~mask</i> , reset value 0x00, 'mask follows' blocks interrupts until the mask word is written); <i>MAX_PIO</i> raised to 3.	Mode-2 operation and correct interrupt masking.
z80sio.c	WR0 'return from interrupt' command handled; register RR2 (interrupt vector) readable; register pointer reset after reads; <i>interrupt_check()</i> only called on a state change.	Correct daisy-chain behaviour; speed.
z80daisy.c/.h	New <i>z80daisy_inquire_irq_state()</i> and <i>z80daisy_debug_irq_state()</i> .	The driver inspects the chain without acknowledging an interrupt.
z80.c	Debugger read-op hook (<i>z80_debug_cpu_readop</i>).	Debugger crashed on the 8-bit board's 'MMU lite' banking. Debug only.

6.2 Core interventions

File	Change	Purpose
memory.c	The macros <i>READBYTE/WRITEBYTE</i> for 16, 32 and 64-bit buses no longer shift the address or select the byte lane; the old versions are commented out with a long <i>FIXME</i> .	Z8001 I/O space needs 16-bit width with byte accesses to odd addresses. The author notes that the stock macros lose address information and caused crashes with -O3.
debug/debugcpu.c, debugger.h	<i>debug_request_update()</i> triggers a refresh of all disassembly views.	Keeps the debugger consistent when the MMU mapping changes.
cpuintf.c, cpuexec.h	<i>CPU_Z8000</i> replaced by <i>CPU_Z8001</i> , <i>HAS_Z8001</i> .	Registers the new CPU.
mame.c	Frame update after soft reset; leaving the machine no longer returns to the game-selection menu; disclaimer text removed.	Behave like a computer with a single machine.

6.3 User interface, input and options

File	Change
ui.c/.h, uimenu.c/.h	Replaceable in-game handler. New menus: floppy selection for drive 0 and 1, TTY selection (tty0 to tty7, language 'Deutsch'), About box, 'save disk on exit' dialog.
emuopts.c/.h	New options <i>floppy_directory</i> (default 'floppies'), <i>floppy_file_0</i> , <i>floppy_file_1</i> .
osd/windows/input.c/.h, window.c	Native keyboard: WM_CHAR and special keys (arrows, Del, Ins, F-keys) are written into a 64-byte buffer read through <i>osd_read_ntv_keyboard()</i> , bypassing MAME's key mapping. Windows only.

6.4 Branding and build files (no emulation impact)

*cli*front.c, mame.h, version.c and osd/windows/verinfo.c change names and texts. *tiny.c* and *tiny.mak* reduce the driver list to *p8000_8* and *p8000_16* and enable the Z8001 and BEEP sound. *cpu.mak*, *sound.mak*, *emu.mak* and *windows.mak* are trimmed to what is needed plus the new entries and *-\$(\$SRC)/imports*. The large diffs in *cpu.mak* (1,173 removed lines) and *sound.mak* (449) are removals of unused CPUs and sound chips, not functional changes.

7. Comparison with current MAME

Reference: mamedev/mame master, commit f26abf2b, 13 October 2026. In the tables below green means present, yellow means to be verified, red means missing.

7.1 Components that exist

P8000EMU component	Current MAME
Z8001 CPU (segmented)	<i>z8001_device</i> in <code>src/devices/cpu/z8000/z8000.cpp</code> , derived from <i>z8002_device</i> , with segmented mode and segment trap.
Z8010 MMU	<code>src/devices/machine/z8010.cpp</code> (498 lines, BSD-3-Clause). Not used by <i>p8k.cpp</i> .
i8272 controller	<i>i8272a_device</i> in <code>upd765.cpp</code> ; already used by <i>p8k.cpp</i> for the 8-bit board.
CopyQM images	<code>src/lib/formats/cqm_dsk.cpp</code> (BSD-3-Clause), read-only.
Z80 CTC, PIO, SIO, DMA, daisy chain	All present as modern devices with their own interrupt handling.
Driver skeleton	<code>src/mame/skeleton/p8k.cpp</code> (575 lines, BSD-3-Clause, Fabio Priuli, September 2009), 'based on Matt Knoth's emulator'. Machines <i>p8k</i> and <i>p8k_16</i> , both flagged <code>MACHINE_NOT_WORKING</code> .

7.2 The original fixes, one by one

Fix in P8000EMU	Finding in current MAME	Assessment
DMA: interrupts, read sequence, status byte, WR6 commands, WR3 mask byte	All visible in <i>z80dma.cpp</i> (initiate read sequence, enable/disable interrupts, <i>trigger_interrupt</i> , <i>if (data & 0x08)</i> for the WR3 mask byte).	Probably included
DMA: byte counting (N+1 bytes)	Different mechanism (<i>m_byte_counter == m_count</i> , a comment 'hack?: count=0 ...'). The source also contains <i>case 0xff: // TODO: p8k triggers this, it probably crashed</i> .	To verify; possible real difference
CTC: counter-mode period, read in reset state	<i>period()</i> and <i>read()</i> handle both cases.	Probably included
CTC: down-counter read-back	Computes <i>remaining / period + 1</i> instead of <i>tconst - elapsed / period</i> . Whether both always agree is open.	To verify; off-by-one possible
CTC: <i>z80ctc_create_interrupt</i>	Does not exist and should not be needed. The clock or trigger input can be driven from the driver.	Re-implement in the driver
PIO: mask logic, 'mask follows', reset value	Reset sets the mask to 0xff and the comparison uses the inverted mask; 'disable interrupts until mask is written' is present.	Probably included
PIO: ready-callback encoding, <code>MAX_PIO</code>	Concern the old API.	Obsolete
SIO: RETI command, RR2	<i>WR0_RETURN_FROM_INT</i> and RR2 vector handling exist. WR2 'vectored interrupt' is marked partially supported.	Probably included; check vector behaviour
SIO/DMA: performance of <i>interrupt_check</i>	Different architecture.	Obsolete
<i>z80daisy_inquire_irq_state</i>	<i>p8k.cpp</i> solves this with its own <i>p8k_16_daisy_device</i> ; its header lists 'properly implement Z80 daisy chain in 16 bit board' as TODO.	Solved differently, still open in the driver
Z80 debugger read-op, <i>debugcpu.c</i> refresh	Modern debugger works on address spaces.	Obsolete
<i>memory.c</i> byte access in I/O space	Modern memory system has per-map widths and masks. The byte-wise odd-address behaviour of the Z8001 must be reproduced in the I/O map of the driver.	Obsolete as a patch; behaviour must be re-created
Z8001 instruction accuracy	Two independent implementations. P8000EMU has its own table (1,212 vs. 596 lines) and 3,697 lines of segmented operations.	To verify with test programs

7.3 What does not exist in current MAME

P8000EMU component	Finding
Z8010 in the 16-bit driver	p8k.cpp uses a fixed memory map (ROM at the bottom, shared RAM above it) and does not reference the MMU device.
8-bit-board abstraction layer (p8000_8_abstr.c)	No counterpart. This is the largest missing piece for a working 16-bit system.
WDC Winchester controller (p8000_16_wdc.c)	No counterpart; a hard-disk image device or a custom controller must be written.
Terminal, keyboard, floppy menus	Modern MAME uses natural keyboard, RS-232/terminal devices and media slots (-flop1/-flop2). The P8000EMU behaviour (TTY 0 to 7, German, save disk on exit) must be mapped to these.
Writing to floppy images	cqm_dsk is read-only in current MAME. How P8000EMU stores changes must be read from cqm_image.c and the exit callbacks before a replacement can be designed.

8. Licensing

- The P8000 drivers, the video code and i8272_floppies are 'freeware for non-commercial purposes'. The author reserves the right to change the terms 'at any time, including retroactively', and modified files must carry a change notice. This is not compatible with the BSD-3-Clause and GPL terms under which MAME is distributed.
- z8010.c is GPLv3, cqm_image.c is LGPL (derived from LIBDSK) and the Z8001 core is based on the MAME/GPL dual-licensed Z8002 core.
- Consequence for a re-implementation in current MAME: use P8000EMU as a source of **hardware behaviour and test knowledge**, but write the code independently. This is also how Fabio Priuli's 2009 skeleton was created.
- If material is ever to be submitted to MAME, asking Matt Knoth for permission or a relicence would be advisable. His 2009 contact address (p8000@knothusa.net) may be outdated. This is not legal advice.

9. Feasibility assessment for stage 3

Overall: feasible, and considerably more so than expected before the comparison, because most devices and a driver skeleton already exist. The work is therefore mostly integration and verification, not writing CPUs or controllers from scratch.

Work package	Effort	Comment
Run p8k and establish what works on the 8-bit board	Small	Needs ROM sets and floppy images from the requester.
Verify DMA counting and CTC read-back against the original behaviour	Small to medium	Small test cases derived from the original z80dma.c and z80ctc.c; fix in the device only if a real difference is proven.
16-bit board: Z8010, memory map with MMU, daisy chain, timer	Medium	Core of the missing functionality; the Z8010 device already exists.
8-bit-board abstraction layer for the 16-bit board	Medium to large	Requires careful reading of p8000_8_abstr.c and the P8000 documentation.
WDC Winchester controller	Medium	Small command set; disk image handling via MAME's hard-disk support.
Terminal, keyboard, floppy handling in the modern style	Medium	Mapping to natural keyboard, terminal device and media slots.
Z8001 edge-case verification	Unknown until tested	Depends on available test software.

Limits and dependencies

- **Build and test.** A full MAME build was not available in the analysis environment, so no code has been compiled or run yet. A reduced build limited to the P8000 driver may be possible and would be tried first; otherwise the requester would build and report results.
- **Test material.** ROMs (monitors, character generator), floppy images and, ideally, known-good software are needed to judge correctness.
- **API accuracy.** The current MAME sources are the authoritative reference for the device API; code is written against them, not from memory.
- **Documentation.** Where P8000EMU encoded hardware behaviour only implicitly, conclusions are drawn from code reading and period documentation and carry some uncertainty.

10. Recommended next steps

#	Step	Result
1	Build and start p8k with the available ROMs.	Baseline for the 8-bit board.
2	Targeted tests of DMA byte counting and CTC down-counter read-back.	Decision whether device fixes are needed.
3	Extend the 16-bit side of p8k .cpp: Z8010 integration, memory map, daisy chain, timer-driven interrupts.	16-bit board boots further than today.
4	Re-create the 8-bit abstraction layer and the WDC controller.	Peripherals for the 16-bit system.
5	Map terminal, keyboard and floppy handling to modern MAME mechanisms.	Usable system without custom UI patches.
6	Verify Z8001 edge cases against P8000EMU behaviour.	Confidence in CPU accuracy.

Appendix A - Modified files (35)

Added and removed lines are counted from the unified diffs (line endings ignored). 'Category' follows sections 6.1 to 6.4.

File	+ lines	- lines	Category
emu/clifront.c	5	2	6.4
emu/cpu/cpu.mak	21	1633	6.4
emu/cpu/z80/z80.c	22	0	6.1
emu/cpu/z80/z80daisy.c	24	0	6.1
emu/cpu/z80/z80daisy.h	2	0	6.1
emu/cpuexec.h	1	1	6.2
emu/cpuintrf.c	3	3	6.2
emu/debug/debugcpu.c	15	0	6.2
emu/debugger.h	2	1	6.2
emu/emu.mak	7	67	6.4
emu/emuopts.c	4	0	6.3
emu/emuopts.h	4	0	6.3
emu/machine/z80ctc.c	21	5	6.1
emu/machine/z80ctc.h	1	2	6.1
emu/machine/z80dma.c	151	26	6.1
emu/machine/z80dma.h	14	8	6.1
emu/machine/z80pio.c	7	6	6.1
emu/machine/z80pio.h	2	1	6.1
emu/machine/z80sio.c	27	5	6.1
emu/mame.c	13	17	6.2
emu/mame.h	1	1	6.4
emu/memory.c	35	18	6.2
emu/sound/sound.mak	4	692	6.4
emu/ui.c	55	4	6.3
emu/ui.h	11	0	6.3
emu/uimenu.c	348	4	6.3
emu/uimenu.h	17	0	6.3
mame/tiny.c	3	16	6.4
mame/tiny.mak	7	42	6.4
osd/windows/input.c	32	0	6.3
osd/windows/input.h	4	0	6.3
osd/windows/verinfo.c	8	8	6.4
osd/windows/window.c	19	6	6.3
osd/windows/windows.mak	3	1	6.4
version.c	1	1	6.4

Appendix B - New files (25)

File	Lines
README	19

File	Lines
emu/cpu/z8001/8001dasm.c	379
emu/cpu/z8001/z8001.c	919
emu/cpu/z8001/z8001.h	84
emu/cpu/z8001/z8001cpu.h	207
emu/cpu/z8001/z8001ops.c	6,562
emu/cpu/z8001/z8001ops_seg.c	3,697
emu/cpu/z8001/z8001tbl.c	1,212
emu/cpu/z8001/z8010.c	259
emu/cpu/z8001/z8010.h	175
emu/machine/cqm_image.c	723
emu/machine/cqm_image.h	42
emu/machine/i8272_floppies.c	582
emu/machine/i8272_floppies.h	43
imports/d3d8.h	1,276
imports/d3d8caps.h	354
imports/d3d8types.h	1,600
imports/dinput.h	4,373
imports/dsound.h	2,345
mame/drivers/p8000_16.c	636
mame/drivers/p8000_16_wdc.c	305
mame/drivers/p8000_8.c	508
mame/drivers/p8000_8_abstr.c	390
mame/includes/p8000_8.h	42
mame/video/p8000_8.c	415

Appendix C - Sources

- Input archives supplied by the requester: mame-mame0123.zip, p8000_mame_src.zip (including the P8000 README).
- Current MAME, github.com/mamedev/mame, master at commit f26abf2b (13 Oct 2026):
src/mame/skeleton/p8k.cpp, src/devices/machine/z8010.cpp, z80dma.cpp, z80ctc.cpp, z80pio.cpp, z80sio.cpp, src/devices/cpu/z8000/z8000.h, src/lib/formats/cqm_dsk.cpp.
- git.redump.net mirror of MAME: src/mame/skeleton/p8k.cpp and src/lib/formats/cqm_dsk.cpp.