

# SQLite hwtime.h 文件嵌入式汇编代码分析报告

石曜铭

2025 年 6 月 11 日

## 1 引言

本文以 SQLite 项目的 `src/hwtime.h` 文件为对象，对其中的嵌入式汇编代码进行转换、注释及分析。

`hwtime.h` 文件的作用是为不同平台（主要是 x86、x86\_64、PowerPC 架构）提供一种高精度计时功能，获取 CPU 级别的高分辨率时钟计数，常用于性能分析和代码剖析。文件内定义了 `sqlite3Hwtime()` 函数，它直接读取 CPU 的时间戳计数器（如 x86 的 RDTSC 指令），返回自处理器启动以来的时钟周期数。这可以用来测量代码片段实际消耗的 CPU 周期，精度远高于普通的系统时间函数。为了适应不同的编译器和 CPU 架构，可以使用内联汇编方便地实现支持。

## 2 x86/x86\_64 架构的实现

### 2.1 GCC 下的 x86/x86\_64 实现

C 代码片段：

```
__inline__ sqlite_uint64 sqlite3Hwtime(void){
    unsigned int lo, hi;
    __asm__ __volatile__ ("rdtsc" : "=a" (lo), "=d" (hi));
    return (sqlite_uint64)hi << 32 | lo;
}
```

转换为汇编 (Intel 语法)：

```
rdtsc                ; 读取时间戳计数器到 EDX:EAX
; EAX -> lo
; EDX -> hi
; (C 端合并为 64 位整数返回)
```

分析与注释：

- `rdtsc` 指令将 CPU 自上电以来的时钟周期数分别放入 EDX:EAX。

- C 代码通过内联汇编提取这两个寄存器的值，组合成 64 位整数，便于高精度计时。
- 用于性能分析和微基准测试。

## 2.2 MSVC 下的 x86 实现

C 代码片段：

```
__declspec(naked) __inline sqlite_uint64 __cdecl sqlite3Hwtime(void){
    __asm {
        rdtsc
        ret
    }
}
```

转换为汇编 (Intel 语法)：

```
rdtsc                ; 读取时间戳计数器到 EDX:EAX
ret                  ; 直接返回，返回值即为 EDX:EAX 组成的 64 位整数
```

分析与注释：

- `__declspec(naked)` 表示该函数无自动生成的函数序言/结尾，直接运行汇编体。
- 与 GCC 版本功能一致，返回硬件周期计数。

## 3 PowerPC 架构的实现

C 代码片段：

```
__inline__ sqlite_uint64 sqlite3Hwtime(void){
    unsigned long long retval;
    unsigned long junk;
    __asm__ __volatile__ ("\n\
    1:mtb1\n\
    mftb1\n\
    mftb1\n\
    mftb1\n\
    cmpw1\n\
    bne1b"
        : "=r" (retval), "=r" (junk));
    return retval;
}
```

转换为汇编 (PowerPC 语法)：

```

1:
    mftbu    rJunk        ; 读取 time base upper (高 32 位) 到 rJunk
    mftb     rRetLow      ; 读取 time base lower (低 32 位) 到 rRetLow
    mftbu    rRetHi       ; 再次读取 time base upper 到 rRetHi
    cmpw     rRetHi, rJunk ; 比较两次高位是否一致
    bne      1b           ; 若不一致, 表明期间发生溢出, 重试
    ; rRetHi:rRetLow 即为完整的 64 位时间基准

```

#### 分析与注释:

- PowerPC 架构的时间基准寄存器 (time base) 分高低两部分读取。
- 为避免低 32 位读取时发生溢出, 采用两次读取高位, 对比防止跨界, 保证原子性。

## 4 无法使用汇编的情形

#### C 代码片段:

```
sqlite_uint64 sqlite3Hwtime(void){ return ((sqlite_uint64)0); }
```

#### 分析:

- 若编译器/平台不支持内联汇编, 则直接返回 0, 不提供高精度计时功能, 保证代码跨平台可编译。
- 该函数一般只用于调试和性能分析, 不影响 SQLite 主功能。

## 5 综合分析

- `hwtime.h` 通过内嵌汇编, 跨平台地实现了硬件级高分辨率计时功能, 便于开发者进行微秒级性能分析。
- 针对主流 CPU 架构 (x86, x86\_64, PowerPC) 分别优化实现, 保证效率与准确性。
- 若平台不支持, 则以安全的空实现兜底, 提升兼容性。

## 6 参考资料

- RDTSC 指令 - Intel® 64 and IA-32 Architectures Software Developer's Manual
- PowerPC Time Base Register
- SQLite 官方源码仓库 <https://github.com/sqlite/sqlite/blob/master/src/hwtime.h>