

Programming Lab 8D

Time of Day

Topics: Replacing division by reciprocal multiplication, replacing multiplication by a sequence of addition, subtraction and shift instructions.

Prerequisite Reading: Chapters 1-8

Revised: March 25, 2025



Click to download
Lab8D-Main.c

Background: Computers keep track of time using an internal hardware counter that starts at zero when the processor is reset and increments at a rate determined by the CPU clock frequency. Software reads a value from the counter and divides it by the number of CPU clock cycles in one millisecond ($1/1000^{\text{th}}$ of a second); this value is then recorded in a variable and used to update the time of day. Computers with an Internet connection initialize their internal date and time of day from a time server¹ and then update it periodically as long as the computer is running. Our processor has no Internet connection, so this lab sets the initial date and time of day according to when the main program was compiled using the built-in C language macros² `__TIME__` and `__DATE__`.

Assignment: The main program will compile and run without writing any assembly. However, your task is to create equivalent replacements in assembly language for the following two functions found in the C main program.

```
uint32_t Time2Msecs(uint32_t hour, uint32_t mins, uint32_t secs) ;
```

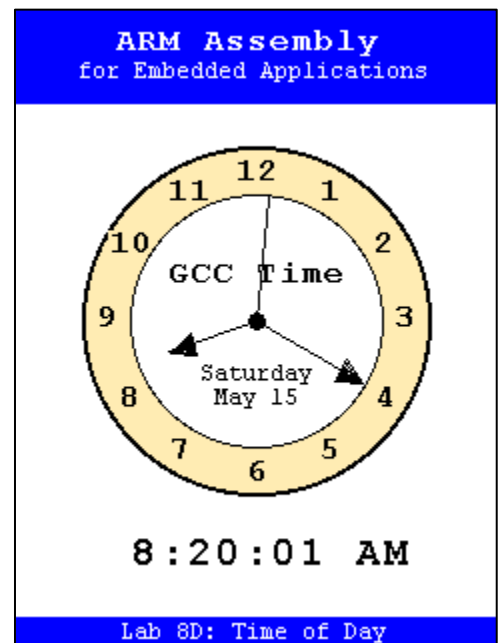
Receives a time measured in hours, minutes and seconds and returns the equivalent total number of milliseconds. Your implementation must not use any multiply instructions (MUL, MLS, etc.). Multiplication must be implemented using a combination of shifting, addition and/or subtraction.

```
void Msecs2Time(uint32_t msec, uint32_t *hour, uint32_t *mins, uint32_t *secs) ;
```

Receives a time measured in milliseconds and stores the equivalent number of hours, minutes and seconds in memory via pointers. Your implementation must not use any divide instructions (SDIV or UDIV). Use [this](#) webpage to find instruction sequences that will divide by a constant without using a divide instruction.

The original C versions of these functions have been defined as “weak” so that the linker will automatically replace them in the executable image by those you create in assembly; you do not need to remove the C versions. This allows you to create and test your assembly language functions one at a time.

Test your code with the main program. If your code is correct, the display should look like the image shown. Errors will be displayed as **white text on a red background**. Every time you run the program, the clock will start at the date and time that the main program was compiled, but will continue to update as long as the program continues to run.



¹ https://en.wikipedia.org/wiki/Time_server

² <https://gcc.gnu.org/onlinedocs/cpp/Standard-Predefined-Macros.html>