

PLASMA Coding Standard

Piotr Luszczyk, Hatem Ltaief, Jakub Kurzak, Bilel Hadri

January 7, 2009

1 FORTRAN Style

Currently PLASMA contains significant amount of FORTRAN code located in the *coreblas/* folder. This code implements single-core operations on matrix tiles. In the future this code might or might not be rewritten to C for portability reasons. The advantage of keeping this code in FORTRAN is its close resemblance of LAPACK code, from which, for the most part, the code was derived. By the same token, the main coding rule applying to development and maintenance of this code is that it should follow LAPACK as closely as possible. This applies to the use of whitespaces, punctuation, indentation, line breaking, the use of lower and uppercase characters, comments, variable naming, etc.

2 C Style

No Trailing Whitespaces: There should be no trailing whitespace characters at the end of lines, no whitespace characters in empty lines and no whitespace characters at the end of files (The last closing curly bracket should be followed by a single newline). This is easy to accomplish by using an editor that shows whitespace characters, such as Kwrite or Kate. Otherwise a *sed* or *awk* "one-liner" script can be used to clean up the file before committing to the repository.

Whitespace Separators: There should be a whitespace between a C language keyword and the left round bracket and a whitespace between the right round bracket and the left curly bracket. There should be no whitespace

immediately after left round bracket and immediately before right round bracket. Comas separating arguments are followed by a single space and not preceded by a space.

Indentation: The unit of indentation is four spaces. the left curly bracket follows the control flow statement in the same line. There is no new-line between the control flow statement and the block enclosed by curly braces. The closing curly bracket is in a new line right after the end of the enclosed block.

There is no specific limit on the length of lines. Up to a 100 columns is fine. Clarity is paramount. For multi-line function calls it is recommended that new lines start in the column immediately following the left bracket.

Tabs: Tab characters should not be used. Tabs should always be emulated by four spaces, a feature available in almost any text editor. If that proves difficult, again, a *sed* or *awk* "one-liner" can be used to do the replacement before the commit.

Variable Declarations: For the most part all variables should be declared at the beginning of each function, unless doing otherwise significantly improves code clarity in a specific case.

Constants: Constants should have appropriate types. If a constant serves as a floating point constant, it should be written with the decimal point. If a constant is a bit mask, it is recommended that it is given in hexadecimal notation.

Printf Strings: ANSI C concatenates strings separated by whitespace. There is no need for multiple printf calls to print a multi-line message. One printf can be used with multiple strings.

F77 Trailing Underscore: When calling a FORTRAN function the trailing underscore should never be used. If the underscore is needed it should be added by an appropriate conditional preprocessor definition in an appropriate header file (e.g.: core-blas.h, lapack.h).

3 Coding Practices

Dead Code: There should be no dead code: no code that is never executed, no including of header files that are not necessary, no unused variables. Dead code can be justified if it serves as a comment, e.g., canonical form of optimized code. In such case the code should be in comments.

OS Interactions: Error checks have to follow each interaction with the OS. The code should never be terminated by the OS. In particular each memory allocation should be checked. The code cannot produce a segmentation fault.

User Interactions: User input needs to be checked for correctness. The user should not be able to cause undefined behavior. In particular the user should not be able to cause termination of the code by the OS.

4 Naming Convention