2nd Italian Workshop on Embedded Systems (IWES 2017)
Computer Science Department - Sapienza University of Rome, Italy
September 7 - 8, 2017

# CC4CS: A Unifying Statement-Level Performance Metric for HW/SW Technologies

Vincenzo Stoico[1], Vittoriano Muttillo[1], Giacomo Valente[1], Luigi Pomante[1], Fausto D'Antonio[2]

[1]Università degli Studi dell'Aquila - Center of Excellence DEWS, L'Aquila, Italy
[2]Thales Alenia Space, Via Campo di Pile, L'Aquila, Italy

In the last thirty years there has been an exponential increase of the spread and evolution of information technologies. In this respect, it is certainly underlined the spiraling of embedded systems. The presence of such systems in everyday life is constant and often almost invisible. Moreover, the adopted design methodology is of critical importance during the development of an embedded system. Unfortunately, such methodologies usually lack generality and can be very effort and time consuming, especially when working at a low level of abstraction. For this reason, working on a higher abstraction level (i.e. *system-level*) is needed and early performance estimation is a fundamental step. One of the most common metric for computer performance analysis is MIPS (*Million Instructions Per Second*) because it is normally available directly on data-sheet. MIPS metrics measures millions of assembly instructions executed per second, and it can be useful for comparing two processors with the same ISA (*Instruction Set Architecture*) but it is pointless in comparing ones with different micro-architectures.

In such a context, the objective of this work is to analyze the usefulness of a metric related to C programming language statements. This kind of metric, called CC4CS (*Clock Cycles for C Statement*), is defined as the ratio between the number of clock cycles required by a target processor to run an application and the number of executed C statements. For this purpose, a framework that helps to calculate this kind of metric for a given set of programs has been realized. Additionally, such a framework is also able to automatically generate large amounts of constrained random inputs and to evaluate statistics on the metric itself. Then, by analyzing the results, it is possible to validate the metric with respect to the performance of several target processors. Summarizing, such a framework allows to easily evaluate CC4CS in a repeatable manner. The working process has been defined by looking at the CC4CS definition. The framework exploits an ISS (*Instruction Set Simulator*) and the simulation permits to calculate the number of clock cycles needed to execute the program while the number of executed C statements is obtained performing a profiling on the host architecture (i.e. the one that executes the framework itself).

Finally, it is worth noting that, since this work avoid reasoning about assembly code related to C statements (i.e. it is based only on C code profiling and target execution time), it can be extended to evaluate CC4CS also for C functions directly implemented in HW by means of HLS (*High Level Synthesis*) techniques. In other words, CC4CS can be used as an early performance metric for HW/SW co-design methodologies (in particular to support system-level timing co-simulations).

## References

[1] V. Stoico, V. Muttillo, G. Valente, F. D'Antonio, L. Pomante, "*CC4CS: A Unifying Statement-Level Performance Metric for HW/SW Technologies*", Euromicro Conference on Digital Systems Design (DSD 2017) - Work in Progress Session, Vienna, Austria, Aug. 30 - Sep. 1, 2017.