Exploring the Limits of LLMs for System-Level Test Program Generation: Can LLaMas Outrun Darwin?

Denis Schwachhofer[1,3], Steffen Becker[1], Stefan Wagner[1,4], Matthias Sauer[2], Ilia Polian[3]
[1]Institute of Software Engineering, University of Stuttgart, Stuttgart, Germany [2]Advantest Europe, Boeblingen, Germany [3]Institute of Computer Engineering and Computer Architecture, University of Stuttgart, Stuttgart, Germany
[4]Technical University of Munich, Heilbronn, Germany

Abstract

System-Level Test (SLT) is important in semiconductor testing as it can detect defects missed by traditional methods. Test engineers use off-the-shelf software to manually compose test suites, often written in high-level languages such as C/C++ or Rust. Several methods for automatically generating test programs have been investigated, using assembly language. However, one could argue that the resulting test programs are not capturing all possible interactions in actual software. Large Language Models (LLMs) can generate code in high-level languages closer to actual software. In this work, we examine the limitations of LLMs and high-level languages for generating SLT programs. We run an experiment using genetic programming (GP) to find an assembly snippet with the highest power consumption. Then, we utilize LLMs to generate C code and demonstrate that the compiler, the enabled optimization level, and the LLM have a significant influence on the resulting power consumption. Furthermore, we show via decompilation that the snippet from the GP run has no direct equivalent in C. Finally, we demonstrate that the initial values have a significant impact on power consumption for both the GP-generated and the decompiled snippet.