Temu: EMUlator Binary Translation

Contemporary emulators used in the European Space industry have been using a technique called interpretation. This method has served us well, as it is easy to implement, and this has resulted in a plethora of different emulators being available for use during flight software development, software validation and operational simulation.

However, interpretation is by definition inefficient and several systems have been designed that utilise a technique called (dynamic) binary translation. Where target code is translated at runtime to host code, and then the host code is executed. The problem with a plain binary translator is that it is not efficient when dealing with code that is executed only once (e.g. boot code), and it is very hard to get right when dealing with odd behaviour such as e.g. branches in delay slots.

This project saw the development of a dynamic binary translator engine augmenting an inhouse developed pre-decoded direct interpreter, resulting in a hybrid system where the interpreter can be used when it is more efficient and the binary translator can be used when code segments have been executed several times.

The developed translation engine is based on the LLVM compiler framework, meaning that it is possible to piggy back on millions of lines of code optimisations developed as part of the open source LLVM framework.