**1 Introduction**

Compilers are some of the most complex software systems, aimed at performing the parsing, transforming and generating executable machine code for a program.

They have two components, a front-end, that includes all the analysis phases, converts the program into an intermediate memory representation; and a back-end, which generates the executable code.

The design and implementation of back-ends which is modular and easily extensible has been important area of research in the last few years.

The Low-Level Virtual Machine (LLVM), a compiler infrastructure is well-suited to the above requirement as it is written in a way that allows the reusability of classes as often as possible.

It also allows components to be shared across different compilers and hence one compiler gets benefited by the improvements made to other.

**1.1** **Goal of Thesis**

In this thesis I have implemented a new backend based on the LLVM framework that generates assembly code for the REISC architecture.

ReISC (Reduced energy Instruction Set Computer) is an 32-bit architecture with support for 8/16/20/32 data size with variable instruction length.

Having support for secure data, fast interrupt response and parallel operations, Reisc architecture targets at the next generation Ultra Low Power and High Performance applications, such as digital signal processing.

**1.2 Organization of this Thesis**