**THUMB Mini Assembler**

**Objective:**

Assembler converts assembly code to the machine code. Input to the assembler is the compiler generated assembly (*.s)* file an output is object (.o) file. This project creates a mini assembler which creates object files for cortex-M architecture profile.

What is included?

Most of the Data processing, Arithmetic, logical, shift operations, load-sore instructions with register, immediate and offset addressing mode from 16-bit THUMB instruction set.

**Steps:**

* Parse each line of the code to determine addressing mode: register/immediate/ register offset/ direct. This step takes care of spaces and tabs.
* Separate out the opcode and operands from instruction. Identify number of operands.
* Choose a correct instruction encoding format from…
  1. Move shifted register
  2. Add/Sub register
  3. Move/add/compare/sub immediate
  4. ALU operations
  5. Load/store
* Get opcode mnemonic from a database file, adjust the register/ immediate values according to the format and generate a 16-bit machine code per line.

What is excluded? (Constraint)

* PUSH/POP instructions, Branch instructions,
* 32-bit ARM instructions

Reference: