Project

Assembly Language Programs

Write a two-pass assembler for the 12 bit accumulator architecture having instructions as follows.

|  |  |  |
| --- | --- | --- |
| Opcode | Meaning | Assembly Opcode |
| 0000 | Clear accumulator | CLA |
| 0001 | Load into accumulator from address | LAC |
| 0010 | Store accumulator contents into address | SAC |
| 0011 | Add address contents to accumulator contents | ADD |
| 0100 | Subtract address contents from accumulator contents | SUB |
| 0101 | Branch to address if accumulator contains zero | BRZ |
| 0110 | Branch to address if accumulator contains negative value | BRN |
| 0111 | Branch to address if accumulator contains positive value | BRP |
| 1000 | Read from terminal and put in address | INP |
| 1001 | Display value in address on terminal | DSP |
| 1010 | Multiply accumulator and address contents | MUL |
| 1011 | Divide accumulator contents by address content. Quotient in R1 and remainder in R2 | DIV |
| 1100 | Stop execution | STP |

There are no Macros and Procs to be assembled. Decide your own error reporting strategy. Assume all operands are declared at the end of instructions, sequentially and in order of their appearance in the code. Each operand occupies one word.

Team size: 2 from your tutorial group only.

Soft copy of code to be submitted for anti-plagiarism check

Grading: Documentation, error reporting, working assembler all carry equal weight.