**TEAM SIXTH SIGMA PROJECT REPORT**

**Team Members:**

# **Illan Buzzetti** – [lxb140230@utdallas.edu](mailto:lxb140230@utdallas.edu)

# **Nirmaan Bajpai** – [nxb140330@utdallas.edu](mailto:nxb140330@utdallas.edu)

# **Thomas Lindholm** – [tml150330@utdallas.edu](mailto:tml150330@utdallas.edu)

# **Daniel Salemzadeh** – [dxs150330@utdallas.edu](mailto:dxs150330@utdallas.edu)

**a) Team member names, tasks done by you and tasks done by each of your teammates. You should document here how every teammate has contributed and especially if you have made extra contributions to the project.**

# **Illan Buzzetti,** lxb140230@utdallas.edu

Providing techniques, tools, examples, guidelines, pseudo instructions to individuals as well as leading the project on side,

Label-Type,

Directives,

Making video,

Assembling the codes (program), debug and project final check point for test

* **Nirmaan Bajpai,** [nxb140330@utdallas.edu](mailto:nxb140330@utdallas.edu)

Handling Documentation,

Implementing the sound code,

Making Project Report,

* **Thomas Lindholm,** [tml150330@utdallas.edu](mailto:tml150330@utdallas.edu)

R-Type,

I-Type,

J-Type,

Branch Type,

Shift Type,

‘lui’ instruction to binary

Open file, Read file, Write to a file, Close file,

* **Daniel Salemzadeh,** [dxs150330@utdallas.edu](mailto:dxs150330@utdallas.edu)

Op code and function code for R-Type,

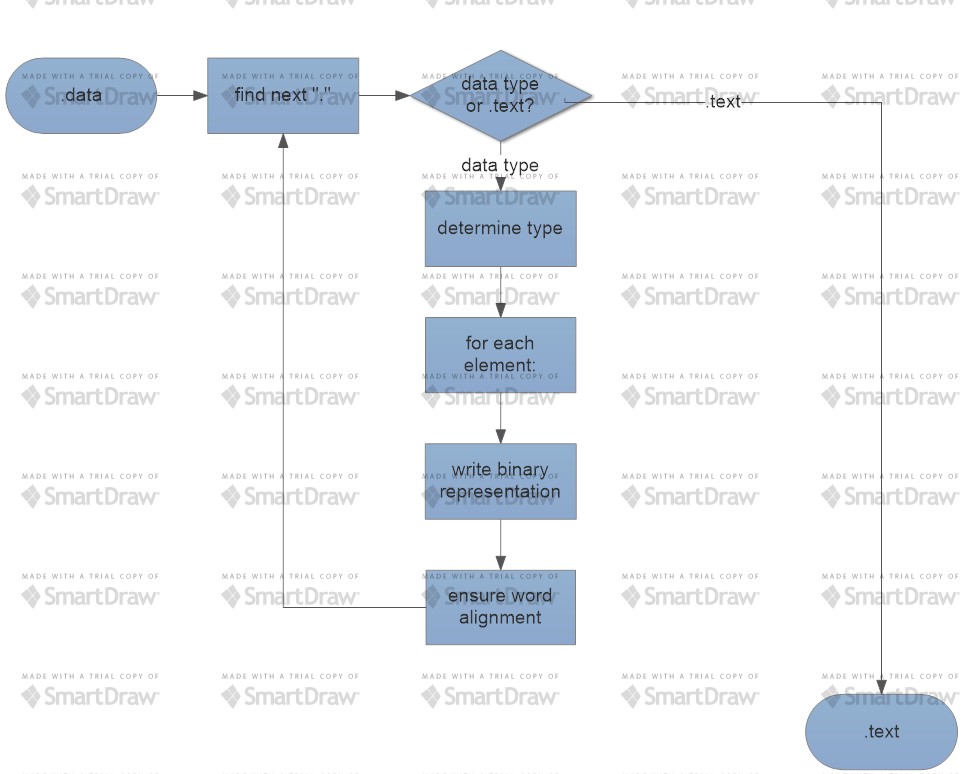
Op code and Immediate for I-Type,

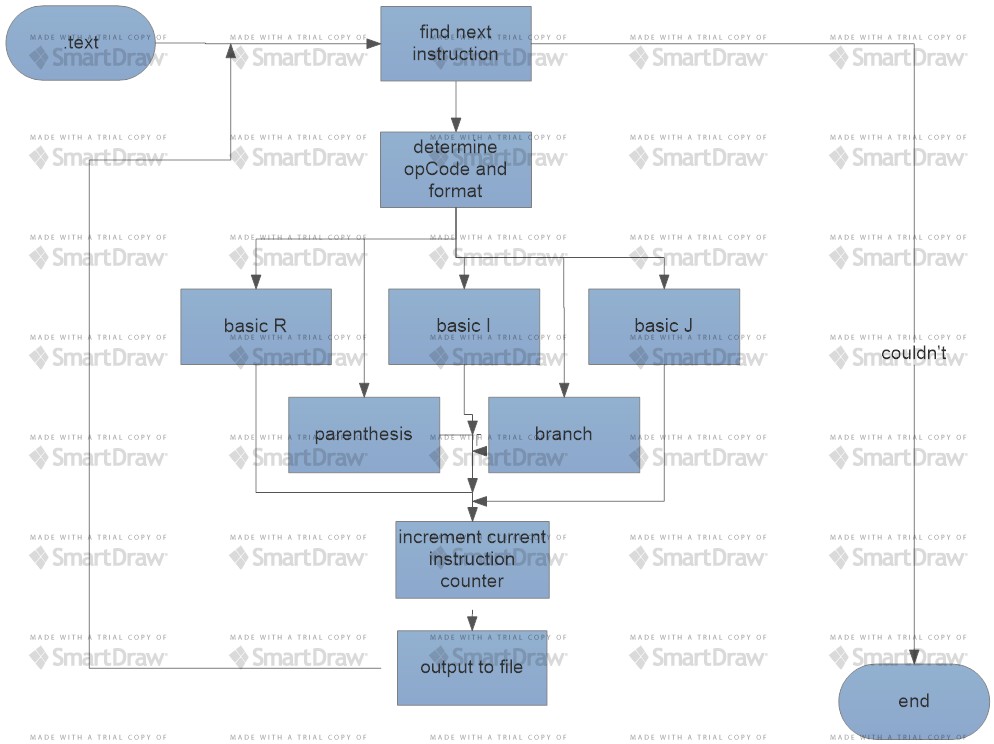
Op code for J-Type

**b) Project Description (the WHAT)**

* Reads a text file (.asm) that contains MIPS assembly languages statements that conform to MIPS assembly language syntax.
* Parses each statement into tokens, e.g. directive, label, operation, operands, comment
* Generates machine code for those MIPS instructions that were syntactically correct into a text segment.
* Allocates memory for those data items into a data segment
* Writes both segments with any other necessary information into an object file in binary format.
* Plays a sound to indicate whether the job was successful or not at the end.

**c. How the program was implemented (the HOW)**

The big picture. Start looks for either .data or .text, upon finding either it goes to the corresponding section in the code. Throughout the program end of file is always checked before reading a new line to guarantee closure at some point.

Within the .data section of the code every line of the file is checked for a directive, upon finding .text the loop exits and gives control to the .text section. Finding any other directive will go to the store function which handles the rest of the line in a manner according to the type of directive found. A 4-byte word aligned buffer is used to write to the text file, this allows for correct storage of strings and a way to ensure word alignment.

Every line from the file is parsed for an instruction. If one isn't found that line is skipped, otherwise the instruction is isolated and passed down the chain of control. The instruction is identified and handled accordingly. The loop exits when the end of file is reached.

The assembler terminates if the textual information doesn't correspond to a sensible instruction coding.

**d) The process the team took to tackle the assignment**

* Individual research and understanding of project requirements.
* Perfect data storage, load and store operations.
* Arithmetic operations and testing.
* The project second check point for test and run.
* Jumping and branching operations and testing.
* The project final check point for test.
* The project written report.
* The project video report.
* The project submission.

**e) Challenges met during the project on how the team overcame them**

* Coordination- Meeting as a group was one of the major challenges we faced during the completion of this project. Not only did our schedules not match but we all were not staying on campus. We solved this easily by meeting after class for half an hour every time and periodically meeting on weekends as well.
* Intellectual- All of us worked together during the completion of this project however some parts were really challenging and required more critical thinking and individual effort. This was handled by Illan and Thomas perfectly as they brainstormed with new ideas to find the most effective way to get the program to compile.
* Learning - As Nirmaan struggled with the programming part, the group encouraged him to get the work done. Guidance from Illan and Daniel helped him cruise through it even though the programming part was tackled by Thomas later. Writing the pseudocode helped him understand the concepts thoroughly. This way we all helped each other by debugging the code, adding useful comments and going through the code line by line.
* Managing Expectations- We used road-mapping through which each member of the group knew about the status of the project, where the project was going and when it will be completed. We eliminated hard dates to relieve stress and replaced it with a less rigid time brackets. This way learning was interesting and exciting at the same time and no one was working under the pressure of time.

**f) What you have learned doing the project.**