**Manchester Baby**

**Coursework Assignment 3**

We are tasked as a group of 5 to develop a program which simulates Manchester Baby. The program should also be capable of converting Assembly code into machine code, which can be then used as code for the simulation.

We approached this problem by taking it one step at a time. We started off by doing as much research about the Manchester Baby as we could. This ensured that we all fully understood what the Manchester Baby was, how it worked and how it was implemented. This gave us a very good idea of how to begin implementation and which parts of the assignment were bigger than others and would need to have more group members working on them. We decided that working in C++ would be the best option for us, since it is what we have most recently been working on in our AC21008 module, and used the g++ compiler in the Linux terminal. With all of this information under our belts, we began to work on implementation.

While implementing the project, we decided to make a GitHub repository to share our work between one another. Through this, we were also able to help one another with debugging certain parts which an individual may have struggled with getting to work.

We had quite a few bugs along the way but fortunately we were able to iron these out and our Manchester Baby met all of the specifications. We had problems with which data structures to use, binary addition and subtraction, loading assembly code, adding blank lines to a vector, and getting rid of comments in assembly code.

When loading assembly code, the vector of vectors of strings that we used to store the code would also store blank lines. To remedy this, we inserted the line received from file into string stream variable, pushed it onto temporary string and then pushed it into a temporary vector of strings. We then checked the size of the vector. If we added blank lines to the vector then the size would equal 0, so we decided to only push the vector onto Assembly Code vector (vector of vectors of string) if the size of vector of strings was greater than 0, and this fixed our problem.

For a while we weren’t sure how to get rid of comments in assembly code, but we noticed that when we pushed the stream string into string, we could check if it contained “;”. therefore, when the program receives “;”, we break out of the loop.

Overall, we are all very happy with the way that our Manchester Baby turned out. We managed to meet all the specifications on time despite having a lot of other deadlines around the same time and even managed a few of the extensions. We would have liked to have implemented a few more extensions however we were unable to do so within the time constrains because of unexpected bugs taking up a lot of our time and most of the group having other deadlines to focus on at the same time as this assignment.

Word Count: