Steven Reeves

CST 250

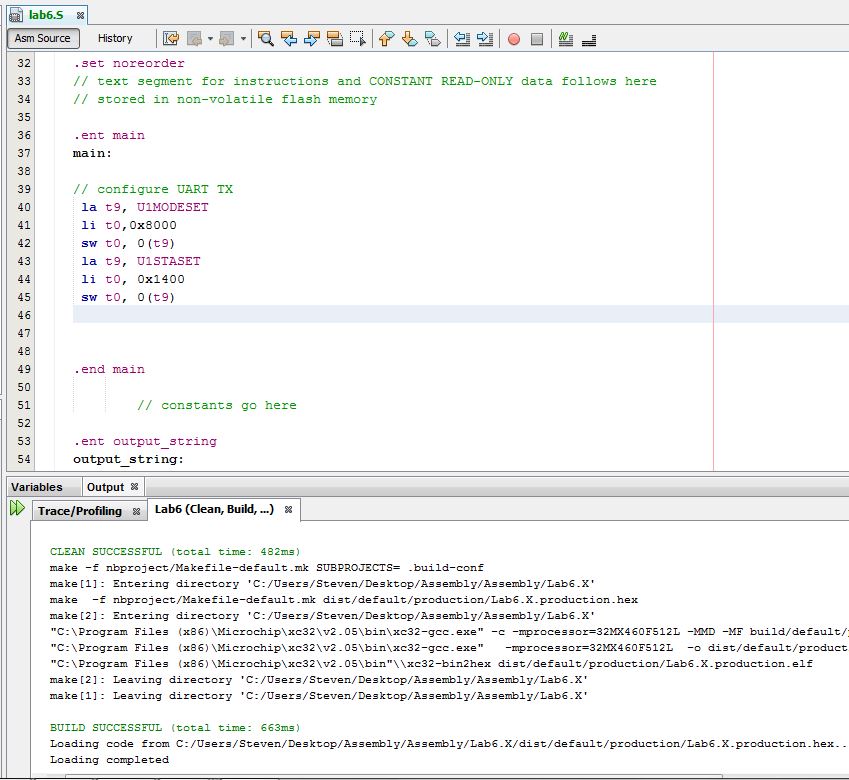
6/2/18

Lab 6 Report

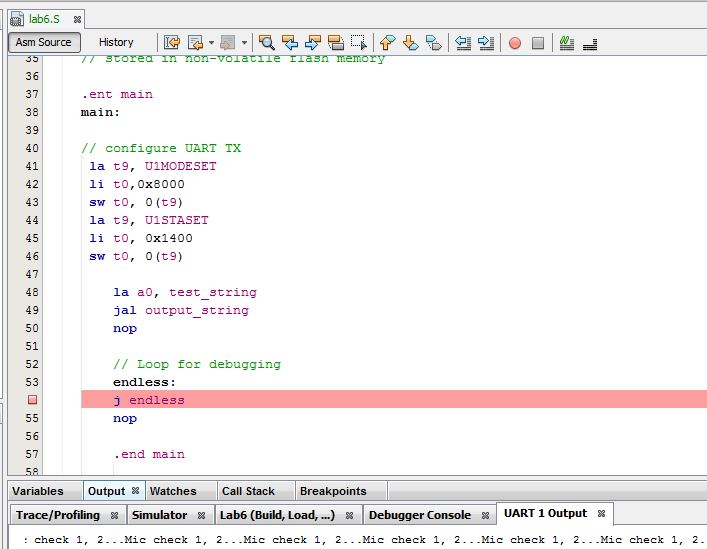
**Introduction:**

The objective of this lab is to replicate the functionality of the square root function. This function will be used to calculate the integer square root of an unsigned 32 bit integer.

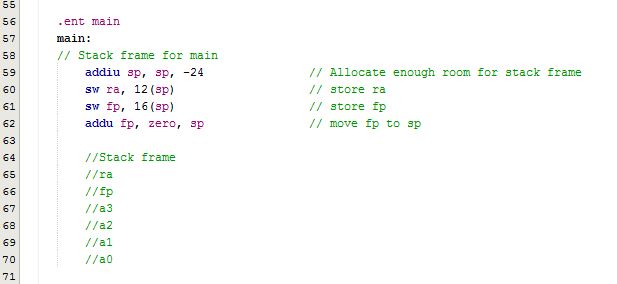
**Part 1:**

The first part of this lab had me set up a UART1 console output and make sure everything builds. I used the setup from Lab 5 and double checked to make sure everything was building correctly. I also included macros for push, pop, and the output\_string function. After following the instructions in Lab 5 for setting up the UART output I tested a build.

I then added a test string to make sure output\_string was working correctly.

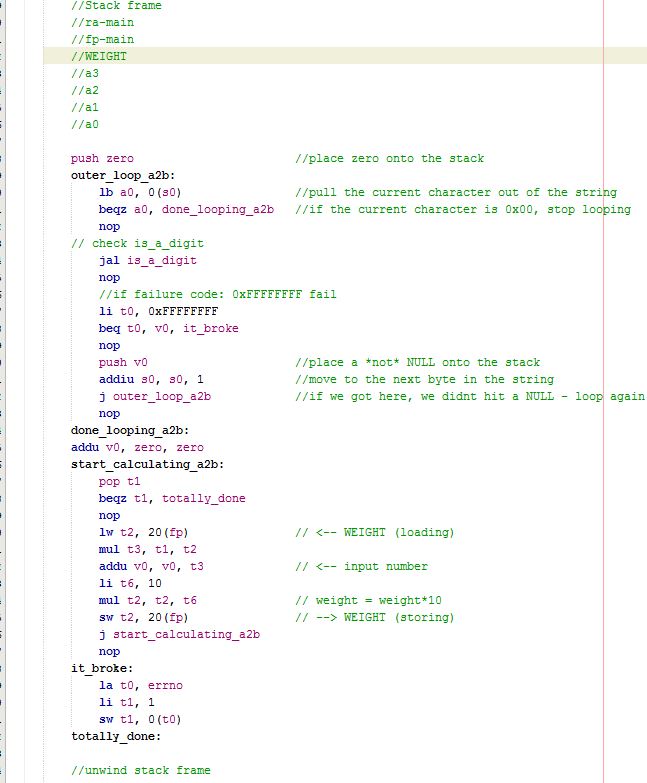


After confirming this worked, I reviewed the description document. After reading the first sentence in that document, I copied the binary\_to\_asciidec function from lab 5 into this project and built to make sure all is well. I then added the call stack functionality from Lab 5 as described in the documentation.

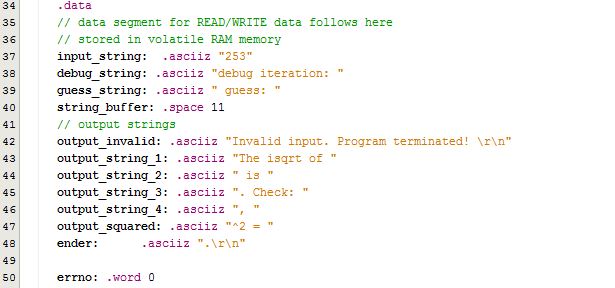
Finally, I added stubs for the needed functions, and checked again that everything built correctly.

**Part 2:**

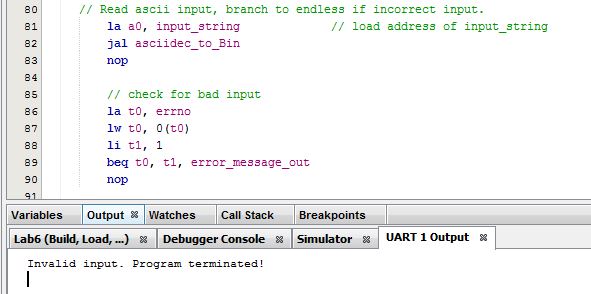
For part 2 of this lab, I followed the algorithm provided in the description document very closely. Upon realizing I needed output to the UART screen to see what answers I was getting, I decided to copy over my is\_a\_digit function over and make sure that was working. The implementation of is\_a\_digit stayed the same from lab 5. After copying this over, I realized that my asciidec\_to\_Bin function needs to call it. After reading through the specifications of this function (and a lot of work in lab) I ended up with the following implementation (stack frame allocation aside):



This is also the place where I decided to declare all the strings I would need to produce the appropriate output.



I then ran the program with invalid input to make sure I got the right error message. This is the output after using the input\_string of “2nope53”. Error\_message\_out just jumps to the output message and unwinding of main’s stack frame.



The basic algorithm provided in the lab description is as follows:

//Basic algorithm

//guess = n / 2;

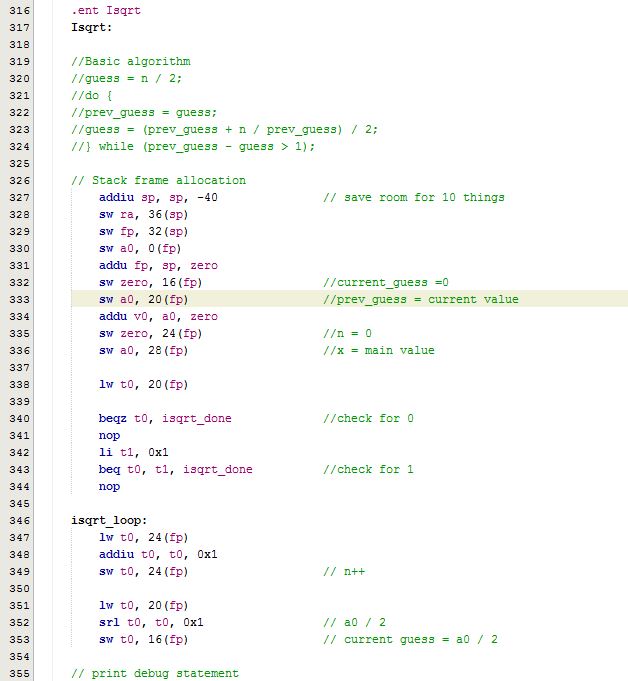
//do {

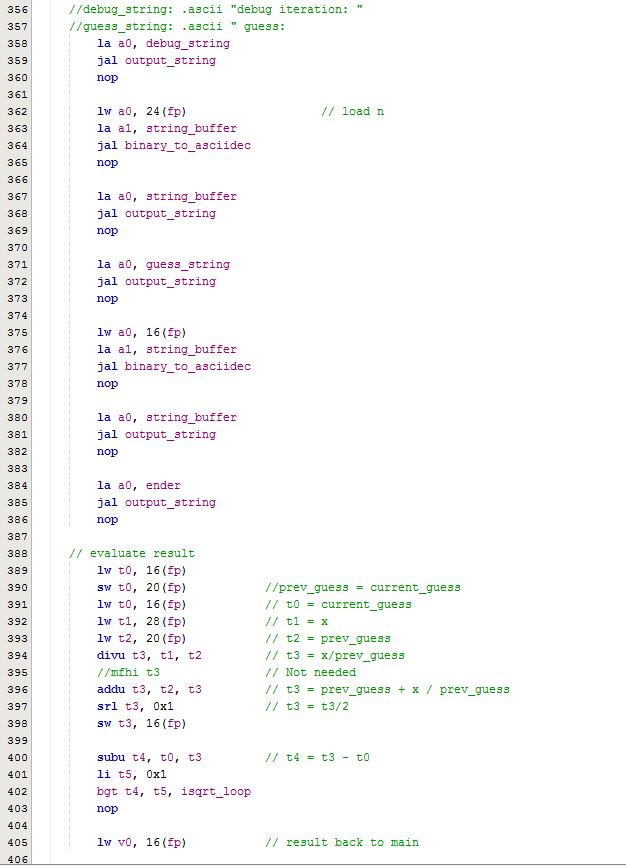
//prev\_guess = guess;

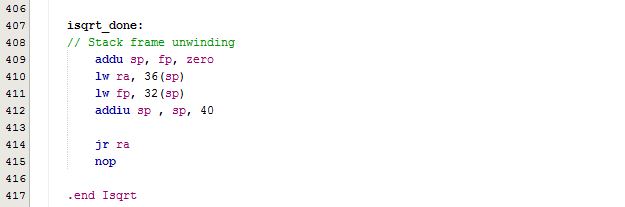
//guess = (prev\_guess + n / prev\_guess) / 2;

//} while (prev\_guess - guess > 1);

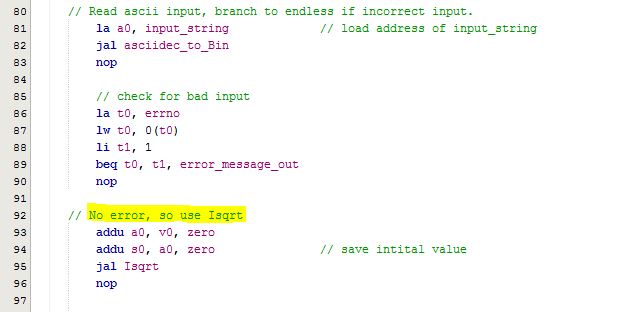
After, again, quite a bit of work in lab the following implementation of isqrt was reached.







After quite a bit of debugging and watching registers, the correct output was verified in main with the following lines added:



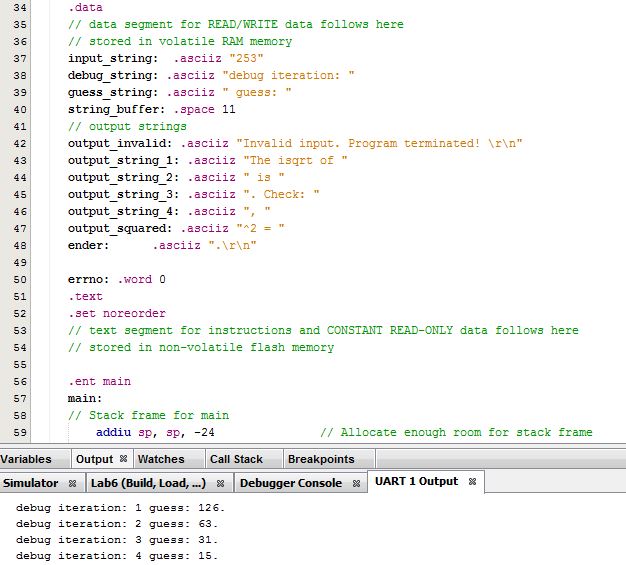
**Part 3:**

Some of part 3 was done in the implementation of isqrt. In particular the requested output of:

debug iteration: 1 guess: 126

…

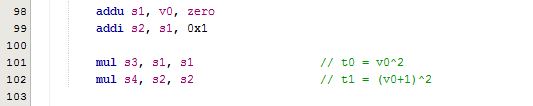
This was confirmed by running a test with the number ‘253’.



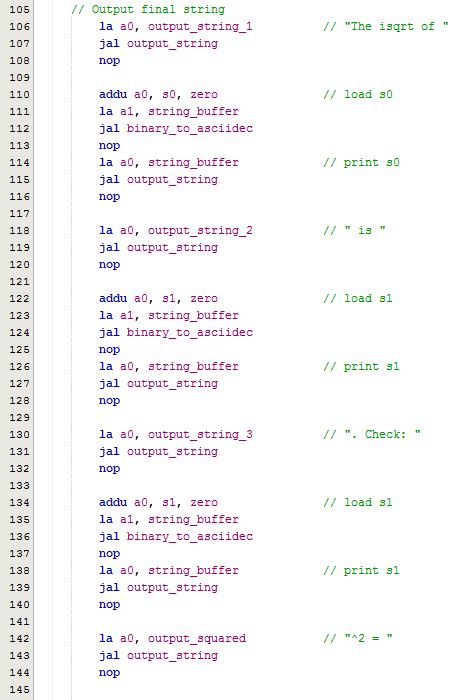
The final part was just to output the last line:

The isqrt of 253 is 15. Check: 15^2 = 225, 16^2 = 256.

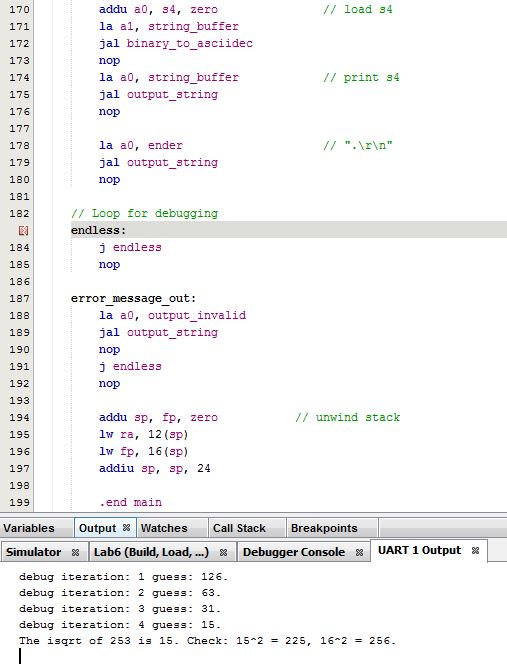
This task proved to be pretty repetitive, as I just had to print out a lot of parts of the string to get everything in order. I did also have to save the result, result +1, and squares of both numbers into s registers.



An example output of the beginning of the last line is shown below:



A final run of the whole build gave me the following output.



**Conclusion:**

In summary, I definitely got a lot of practice allocating/unwinding stack frames, jumping to functions, calling a function from a function, printing to a UART window, and debugging problems. The biggest problem I had in this lab was starting implementation of algorithms. Once I had a basic understanding of how to start the implementation, I was much better off. Working in a group actually helped this a lot. Lab sessions were very valuable in the sense that we could all bounce ideas off each other and catch logic errors early. I did also notice that my output was a bit different than the lab documentation. This is something we all noticed in lab, and were a bit confused about how our number of guesses was off by a bit. Again, working in a group helped us all catch this issue too.