Elaborate on your source program

* What kind of register should we use for MMIO?
  + **The registers used are the receiver control, receiver data**
* What part of the registers did you use and why? What value did you set or get from the register?
  + **Mips can handle memory between 0x00000000 and 0xFFFFFFFF. The memory mapped data goes in the memory mapped IO segment or 0xFFFF0010**
* Show your abstraction expression to calculate the decimal value from the inputs.

**The logic works as such:**

* Load the ‘num’ variable into $t9 containing 3 for counting loop
* Load the x variable that’s initially set to 100 for the multiplication
* Set $s1 to 0 before beginning to remove garbage data
* Make sure receiver control is ready (has 1 in LSB) if it has 0 it loops until ready.
* Read a character from I/O
* Subtract ascii code for 0 from the character. Example: 3-0 is 3 so it would have a 3 ready
* Then it subtracts 1 from the counter that is set to 3 from the variable ‘num’ and put in $t9
* $t8 holds the variable x which is initially set to 100
* It multiplies the entered ascii value by 100 to get the hundredths place
* Then it adds the value in $s0 with the current $s1’s 0 value
* Finally, it divides the value in $t8 by 10
  + **There are 3 loops so $t8’s value goes 100>10>1**
* It does this loop 3 times to recreate the 3 digits entered after passing them through Memory-mapped I/O.
* The final piece gets the final sum then prints the values to the screen.
* Then when done, it exits the program.