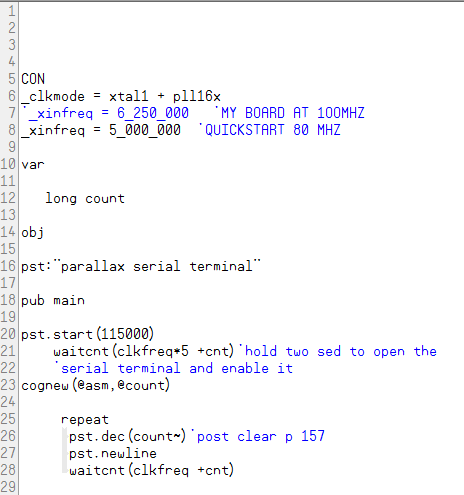
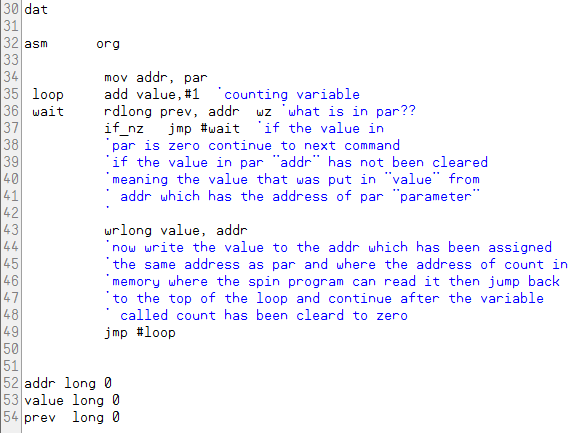
At this point we should be able to get in and out of PASM and do some math and create and target specific array cells.

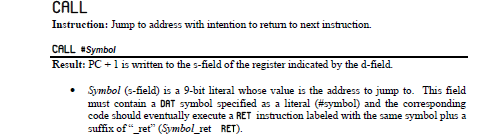
We are now going to revisit those objects and create subroutines with each one.

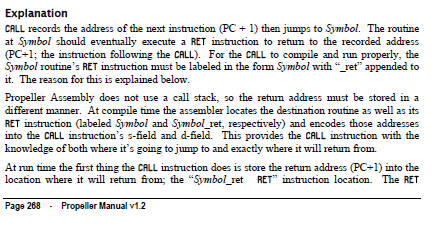
Let’s start with the counting program that counts up from zero.

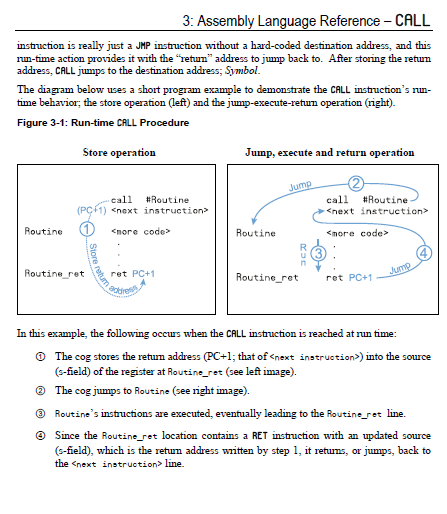


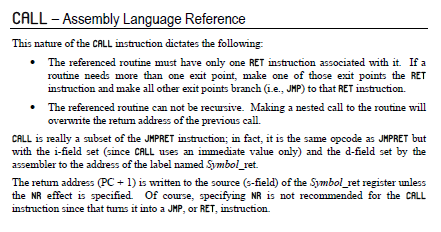


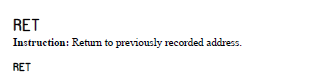
Now we are going to add three lines of code, the code definitions are as follows as seen on lines 44,45 and 61 on the next listing:

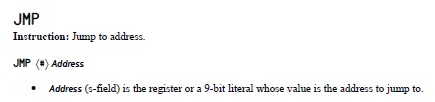


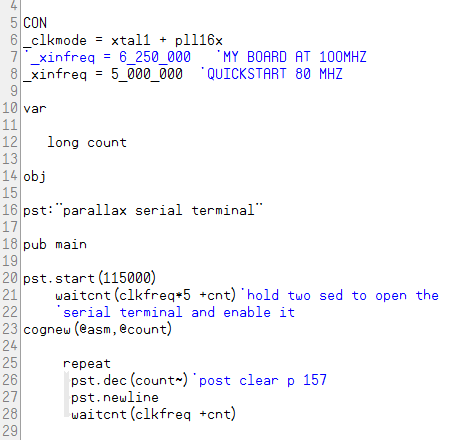


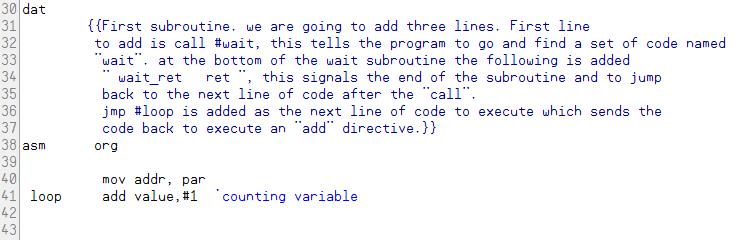


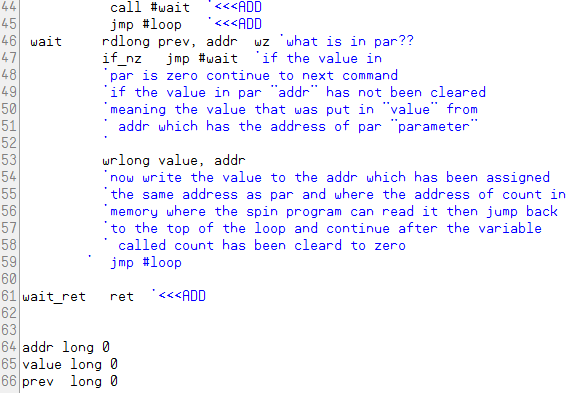








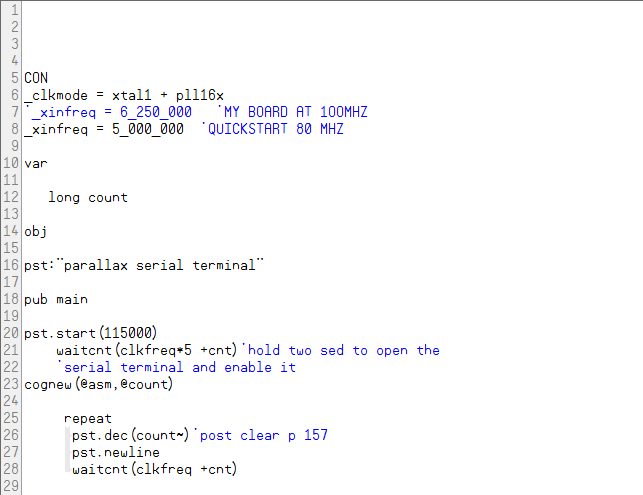


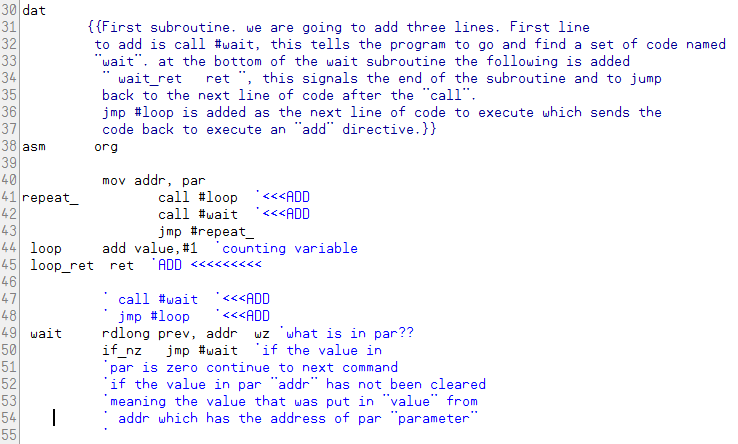


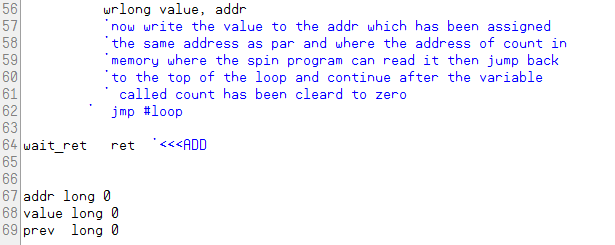
Adding line 44 will call the subroutine named “wait”. The routine will execute the code that is listed there. Upon completion of the code routine the “ret” command will send the code back to the next line of code after the “call” in this case it is a “jmp” meaning a jump to the address listed in the jmp command, which in this case is “loop” which is where the “add” command will add 1 to the value. You should see this:



Now lets get a little deeper and make a couple of other changes. The above code will be modified and will have two subroutines.







We now have two subroutines. The first called is the adding routine that increments the value. The second goes to the wait routine that keeps PASM in lockstep with SPIN both are in a nested loop that loops forever.

You should see this: