

Set a breakpoint at line 14 to see each variable’s address loaded into its resepctive registers.

A screenshot of a computer

Description automatically generated

Set a breakpoint at line 16 to see the value of dot loaded into register X10

.

A screenshot of a computer

Description automatically generated

Set a breakpoint at line 21 to see the registers where the addition and multiplication occurred for the first numbers of the vectors.

A screenshot of a computer

Description automatically generated

Set a breakpoint at line 26 to see the registers where the addition and multiplication occurred for the second numbers of the vectors.

A screenshot of a computer

Description automatically generated

Set a breakpoint at line 31 to see the registers where the addition and multiplication occurred for the third numbers of the vectors.

A screenshot of a computer

Description automatically generated

Use c command to continue the program and stop at line 14 breakpoint. From here we could see the addresses of dot, vec1, and vec2 in registers X0, X1, X2.

A screenshot of a computer

Description automatically generated

Use c command to continue the program and stop at line 16 breakpoint. From here we could see the value of dot, which is 0, loaded into X10.

A screenshot of a computer

Description automatically generated

Use c command to continue the program and stop at line 21 breakpoint. From here we could see the first numbers of each vector in registers X3 and X4. We then multiply them together and add them into dot, giving X10 a value of 10.

A screenshot of a computer

Description automatically generated

Use c command to continue the program and stop at line 26 breakpoint. From here we could see the second numbers of each vector in registers X3 and X4. We then multiply them together and add them into dot, giving X10 a value of 50.

A screenshot of a computer

Description automatically generated

Use c command to continue the program and stop at line 31 breakpoint. From here we could see the second numbers of each vector in registers X3 and X4. We then multiply them together and add them into dot, giving X10 a value of 140.

A computer screen shot of a computer

Description automatically generated

Use s command to step to the next line of the program. From here we can verify that the correct value of dot in X10, which is 140, is stored in the correct memory address of dot.

A computer screen shot of a computer

Description automatically generated

Use *x/1wd 4260144* to view the value in decimal format of the memory address that is saved in X0. This should be the value of dot which is correct since it is 140. This is because 1(10) + 2(20) +3(30) = 140.

A screenshot of a computer

Description automatically generated

Lastly, I then use c to continue the program to the end and have it terminate.