

Laborationsrapport (lab 4)

Gustav Larsson(Guslap-5)
Carl Frisenstam(Carfri-5)

We have learned some C syntax, how to translate assembly code into C. We've learned some about optimization.

The hardest part was to make the C code work, we had some issues with unsigned variables not working as we thought they would.

When comparing the the c generated assembly file with the code from lab 2 there are noticable difference. When it comes to calling convention it seems to be working in the same way as it does in the lab2 code. It uses ax registers to pass arguments into functions, much like the calling convention says. It also stores return values in the vx register as the calling conventions states. A big difference is the complete lack of tx register usage. The assembly code generated from the .c file don't use temporary registers at all. Instead everything is temporarily stored in a vx register before being pushed onto the stack. When storing and loading to and from the stack it uses a frame pointer stored in s8 as an offset.

Clock cycles lab2:	42 226
Clock cycles lab4, not optimized:	109 066
Clock cycles lab4, optimized 1:	28 870
Clock cycles lab4, optimized 2:	28 251
Clock cycles lab4, optimized 3:	27 364

Optimization 3 uses tx registers as opposed to 1 and 2 which stores everything in vx and sx registers. None of the Optimization levels uses a frame pointer when storing items on the stack. O_2 doesn't branch to decode in the if statement as opposed to O_1 and O_3. O_3 doesn't do anything in the else statement of decode. There is a lot of structural reordering between the different optimization levels, code that are in one place in O_2 for example is in a different place in O_3.