Participants:

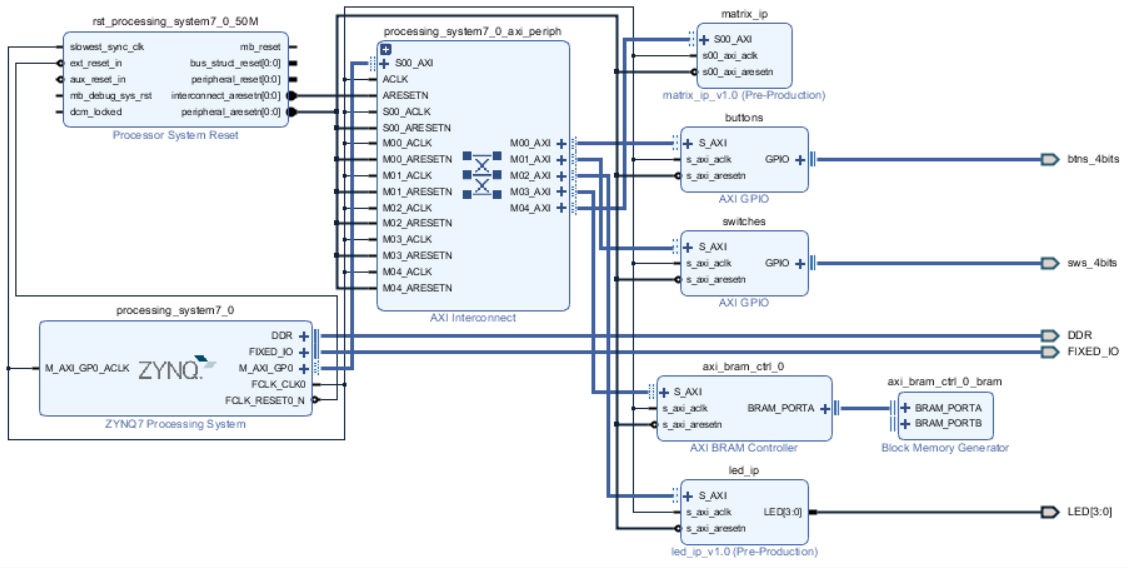
Jacob Munkholm Hansen, 20140479

Jens Jakob Mikkelsen, 201506215

Yang Hanshou, 201902791

**Embedded Real Time Systems – Assignment 2**

For the exercises, the following design will be used:



The buttons and switches will be used in exercise 3 and 4, and the matrix\_ip will be used in exercise 5. The BRAM and led\_ip persists in the design created in the pre-exercises.

# 3) Create a Console Application

The main-program looks like this:

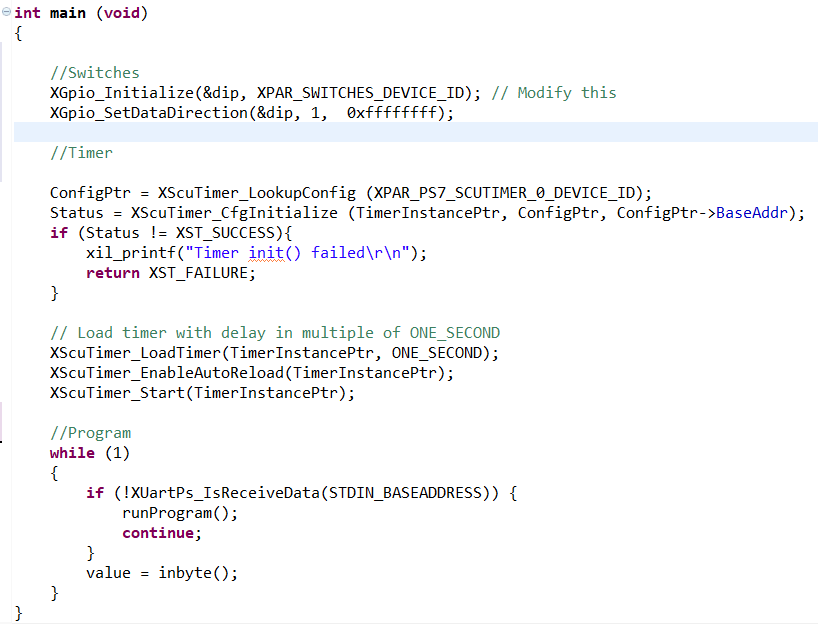


Figure - main program

The main function can be split into two parts; initialization of hardware components (Led’s and timer) and an always-running loop. Every iteration of the loop checks whether data has been received from the uart. If a new value is received, the ‘value’ will be set to it, allowing for the user to switch between the different parts of the program. This functionality is created in the runProgram() function:

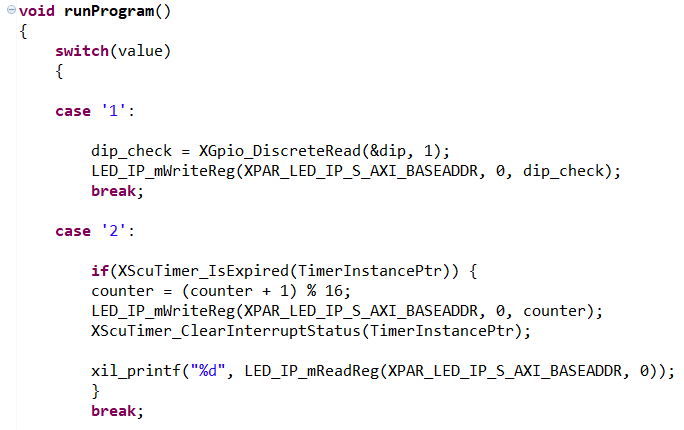


Figure 2 - runProgram() function

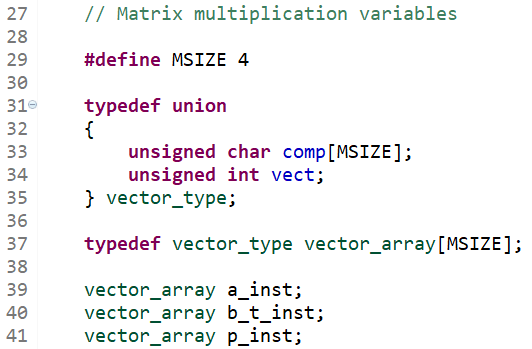
So every time the user writes from the UART, the program will check whether the value corresponds to the case values, and switch between them.

If the value is 1, the switches will be read, and the LED’s will be written to based on what switches are turned on/off.

If the value is 2, every time the timer expires the counter will increase up to 15 and then reset. The LED registers will then be written to, based on the value of the counter. As seen in figure 1, the timer is initialized with a value of one second (325000000 cycles), so it counts every second.

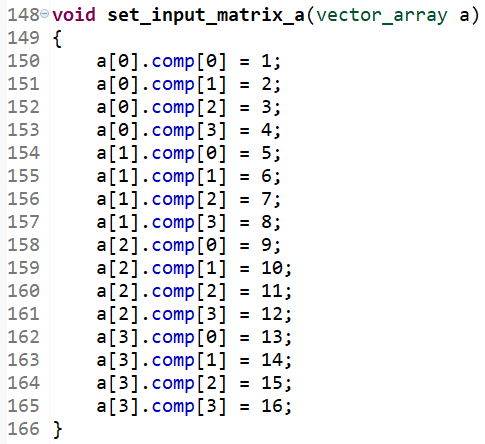
# 4) Create a Matrix Multiplication

## Exercise 4

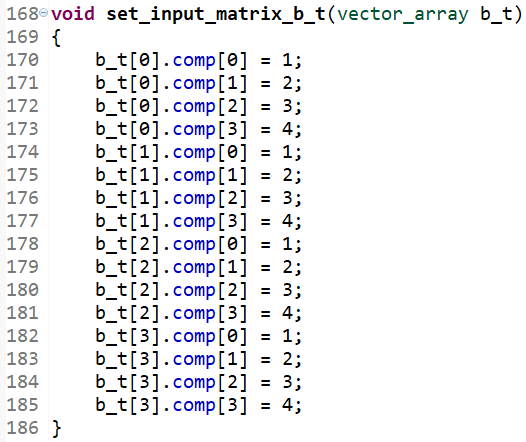


“set\_input\_matrices(…)” has been split into two separate functions “set\_input\_matrix\_a(…)” and “set\_input\_matrix\_b\_t(…)”.

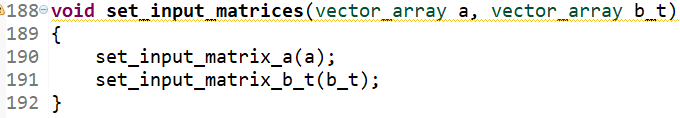
Matrix a is set so that it reflects the matrix in the guidelines:



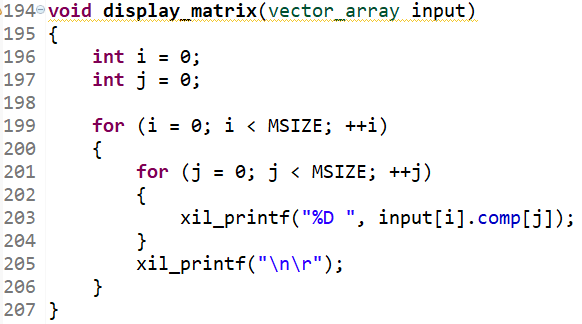
The same is done for matrix b\_t. Notice, however, that since b\_t is b transposed, the rows from the matrix in the guidelines corresponds to the columns of b\_t, so that all values of comp[0] are set to 1, comp[1] to 2, etc.:

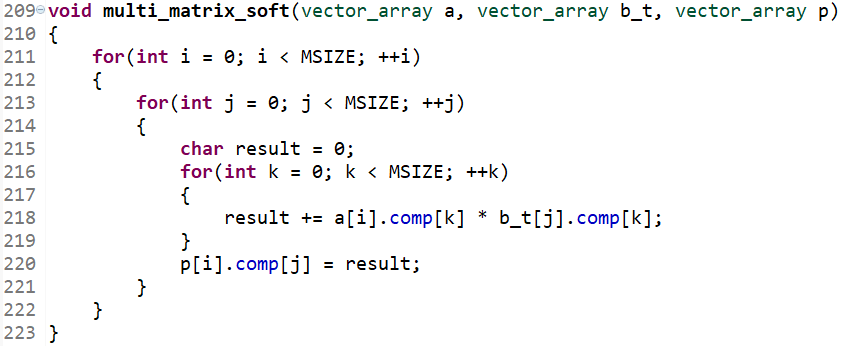


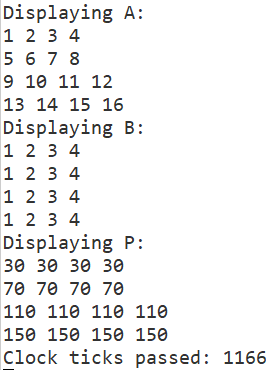
The two functions are then called inside ”set\_input\_matrices(…)”:



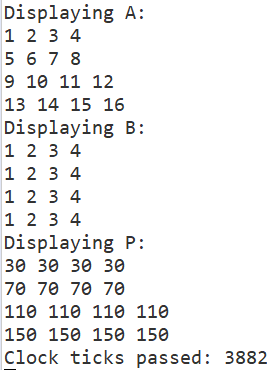
To display a matrix, the function “display\_matrix(…)” has been implemented to loop through each row “i” and each row “j” and printing the value of each row-column element to the UART by calling “xil\_printf(…)”:







# 5) Create a Hardware IP Acceleration



# 7) HLS exercise with SystemC