Programmable Embedded Systems Assignment 2

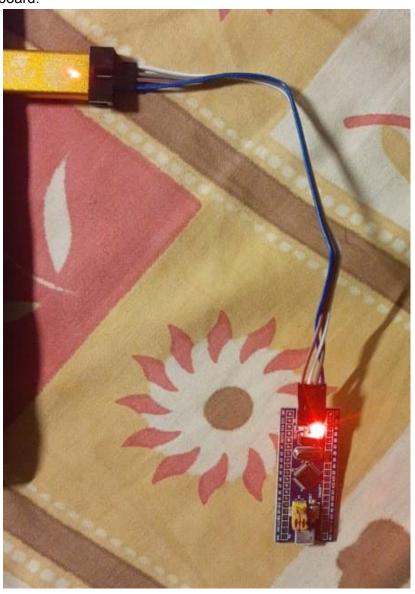
Submitted by Pratyush Jaiswal 18EE35014

Hardware STM32 based implementation of FIR filter

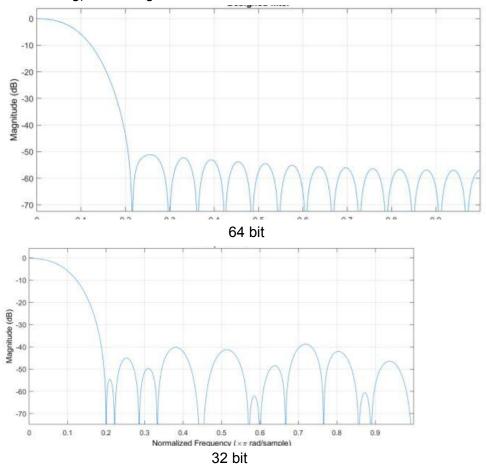
> The code screenshots have not been attached as all the codes were provided by Mam herself.

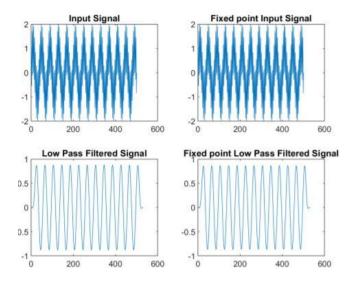
Setup

Connect debugger to laptops and female-female jumpers from the debugger to our STM board



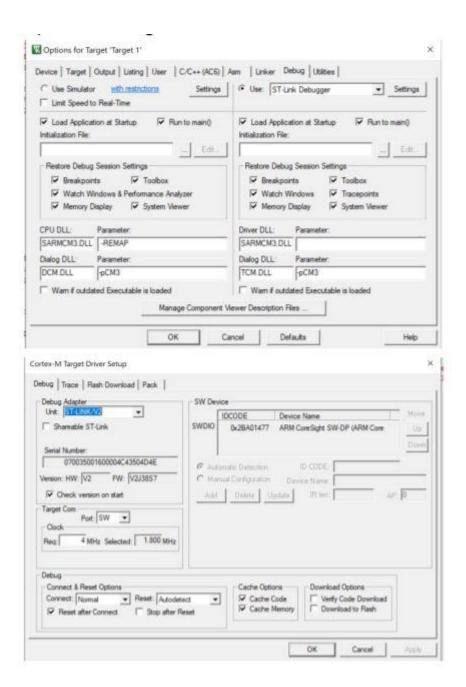
Now running the code in the file named FilterDesignMatlab.m to get our input, filter coefficients and expected output of the filter along with the gain frequency plot(for both 64 and 32-bit filtering). The images are attached below





Input vs output for both types of filtering

The next step is to take those filter coefficients and input signal onto Keil Uvision and write a fir filter code and use our connected stm32 as the debugger instead of the simulator Options for target screenshots attached



The device name is being shown so that means that Keil is able to detect the connected STM and we can run our filter through it

On running the filter the following addresses are stored in the registers

```
りで 中中 自自自 建学 旧品 通
                                                                    1 3 1 a 0 0 a - 1 - 4
* D B B B B P - D - B - 27 - D - B - 27 -
           g 🔲 Disassembly
                                      b __exit_outer_loop ;convolution end
                         42: mul /multiply two 14bit fixed point numbers in r11 4 r12, product in r10, result format A(8,8)
                     add rf, 25, $2 /next filter coeff address
b _inner_loop /repest inner_loop
                            __exit_inner_loop
add r0, r0, r0 /r0=ky(1), address pointer
atth r0, r00 /retore y(1) in ky(1)
r1drah r10, (r0) /r0 cheek y(1)
add r0, r0, t0 /rest input signal address
b_outer_loop /repeat outer_loop
                        59 exit outer loop | 60 b exit outer loop | convolution end
                             _mul :multiply two libit fixed point numbers in ril 4 ri2, product in rio. result format A(S,0)
                                mai rio, ril, ril
ast rio, #f (multiply with 2"-f, f=0, as fixed point multiplication
mp rio, #0
                                  bmi _neg
movt rio,#0m6000 :Extend upper libits if no is positive
                                   movt sil, #inffff :Extend upper lebits if no is negative
                   4
                                                                              9 🔯 Call Stack + Locals
```

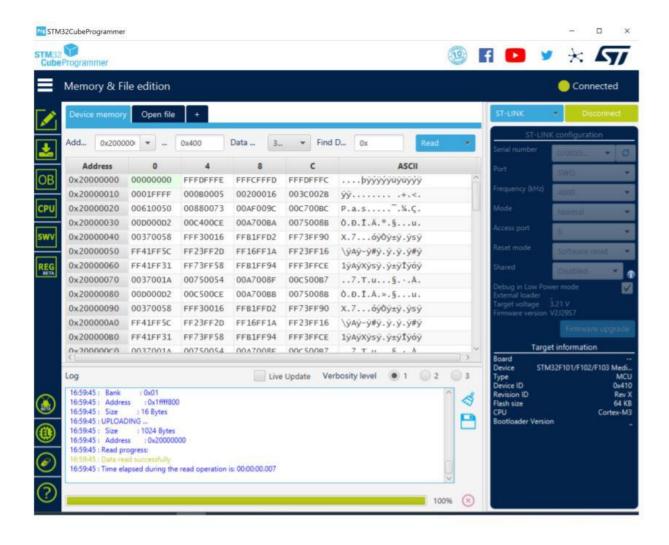
R0 has the starting address of the input signal

R1 has the address of the filter coefficients

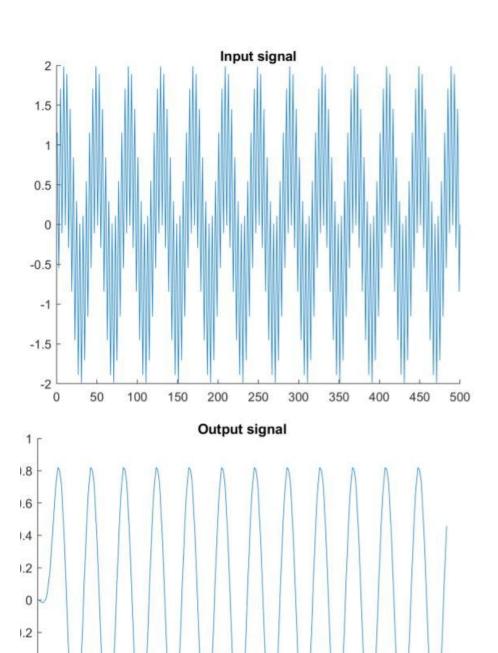
R2 has the starting address of the output

R8 has the ending address of the output

Now, save .bin files for input and output using cube programmer software, example screenshot for output attached



Save bin files and then run the file matlab_reading_stm32.s for reading the input.bin and output.bin files and plot them
The plots are attached below



1.4

1.6

8.

-1

Hence it can be observed that our bin files stored the proper info, meaning our filter ran perfectly