

Part 1:

Profile lab3_fir (using gprof -pg)

Board: Jetson

Compilation option O0

Output:

```
[wes-237b@ubuntu:~/lab/lab3_pynq$ sudo gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 %   cumulative   self           self      total
time  seconds    seconds   calls   ms/call  ms/call  name
50.17    0.02    0.02         1    20.07    20.07  fir(float*, float*, float*, int, int)
25.08    0.03    0.01         1    10.03    10.03  fir_opt(float*, float*, float*, int, int)
25.08    0.04    0.01         1    10.03    10.03  fir_neon(float*, float*, float*, int, int)
 0.00    0.04    0.00         1     0.00     0.00  _GLOBAL__sub_I_Z3firPfS_S_ii
 0.00    0.04    0.00         1     0.00     0.00  _GLOBAL__sub_I_main
 0.00    0.04    0.00         1     0.00     0.00  designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 2 byte(s) for 24.92% of 0.04 seconds

index % time    self  children    called    name
-----
[1]  100.0    0.00    0.04         1/1    <spontaneous>
      0.02    0.00         1/1    main [1]
      0.01    0.00         1/1    fir(float*, float*, float*, int, int) [2]
      0.01    0.00         1/1    fir_opt(float*, float*, float*, int, int) [3]
      0.00    0.00         1/1    fir_neon(float*, float*, float*, int, int) [4]
      0.00    0.00         1/1    designLPF(float*, int, float, float) [14]
-----
[2]   50.0    0.02    0.00         1/1    main [1]
      0.02    0.00         1    fir(float*, float*, float*, int, int) [2]
-----
[3]   25.0    0.01    0.00         1/1    main [1]
      0.01    0.00         1    fir_opt(float*, float*, float*, int, int) [3]
-----
[4]   25.0    0.01    0.00         1/1    main [1]
      0.01    0.00         1    fir_neon(float*, float*, float*, int, int) [4]
-----
[12]  0.0     0.00    0.00         1/1    __libc_csu_init [20]
      0.00    0.00         1    _GLOBAL__sub_I_Z3firPfS_S_ii [12]
-----
[13]  0.0     0.00    0.00         1/1    __libc_csu_init [20]
      0.00    0.00         1    _GLOBAL__sub_I_main [13]
-----
[14]  0.0     0.00    0.00         1/1    main [1]
      0.00    0.00         1    designLPF(float*, int, float, float) [14]
-----

Index by function name

[12] _GLOBAL__sub_I_Z3firPfS_S_ii [2] fir(float*, float*, float*, int, int) [4] fir_neon(float*, float*, float*, int, int)
[13] _GLOBAL__sub_I_main [3] fir_opt(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
wes-237b@ubuntu:~/lab/lab3_pynq$ 0
```

Compilation option O1

```

[wes-237b@ubuntu:~/lab/lab3_pynq$ sudo gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 %   cumulative   self           self      total
time  seconds    seconds   calls   ms/call  ms/call  name
50.17    0.02    0.02         1    20.07    20.07  fir(float*, float*, float*, int, int)
25.08    0.03    0.01         1    10.03    10.03  fir_neon(float*, float*, float*, int, int)
12.54    0.04    0.01         1     5.02     5.02  fir_opt(float*, float*, float*, int, int)
 0.00    0.04    0.00         1     0.00     0.00  _GLOBAL__sub_I_Z3firPfS_S_ii
 0.00    0.04    0.00         1     0.00     0.00  _GLOBAL__sub_I_main
 0.00    0.04    0.00         1     0.00     0.00  designLPF(float*, int, float, float)

      Call graph

granularity: each sample hit covers 2 byte(s) for 28.48% of 0.04 seconds

index % time    self  children   called    name
-----
[1]   100.0    0.00    0.04       1/1    <spontaneous>
      0.02    0.00       1/1    main [1]
      0.01    0.00       1/1    fir(float*, float*, float*, int, int) [2]
      0.01    0.00       1/1    fir_neon(float*, float*, float*, int, int) [3]
      0.00    0.00       1/1    fir_opt(float*, float*, float*, int, int) [4]
      0.00    0.00       1/1    designLPF(float*, int, float, float) [14]
-----
[2]    57.1    0.02    0.00       1/1    main [1]
      0.02    0.00       1/1    fir(float*, float*, float*, int, int) [2]
-----
[3]    28.6    0.01    0.00       1/1    main [1]
      0.01    0.00       1/1    fir_neon(float*, float*, float*, int, int) [3]
-----
[4]    14.3    0.01    0.00       1/1    main [1]
      0.01    0.00       1/1    fir_opt(float*, float*, float*, int, int) [4]
-----
[12]   0.0     0.00    0.00       1/1    __libc_csu_init [20]
      0.00    0.00       1/1    _GLOBAL__sub_I_Z3firPfS_S_ii [12]
-----
[13]   0.0     0.00    0.00       1/1    __libc_csu_init [20]
      0.00    0.00       1/1    _GLOBAL__sub_I_main [13]
-----
[14]   0.0     0.00    0.00       1/1    main [1]
      0.00    0.00       1/1    designLPF(float*, int, float, float) [14]
-----

Index by function name
[12] _GLOBAL__sub_I_Z3firPfS_S_ii [2] fir(float*, float*, float*, int, int) [3] fir_neon(float*, float*, float*, int, int)
[13] _GLOBAL__sub_I_main [4] fir_opt(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
wes-237b@ubuntu:~/lab/lab3_pynq$

```

Compilation option O2

```

[wes-237b@ubuntu:~/lab/lab3_pynq$ sudo gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 %   cumulative   self           self       total
time  seconds    seconds   calls   ms/call  ms/call  name
66.89    0.02    0.02         1    20.07    20.07  fir(float*, float*, float*, int, int)
33.45    0.03    0.01         1    10.03    10.03  fir_opt(float*, float*, float*, int, int)
0.00    0.03    0.00         1     0.00     0.00  _GLOBAL__sub_I_Z3firPfs_S_ii
0.00    0.03    0.00         1     0.00     0.00  _GLOBAL__sub_I_main
0.00    0.03    0.00         1     0.00     0.00  fir_neon(float*, float*, float*, int, int)
0.00    0.03    0.00         1     0.00     0.00  designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 2 byte(s) for 33.22% of 0.03 seconds

index % time    self  children   called    name
-----
[1]   100.0    0.00    0.03       1/1    <spontaneous>
      0.02    0.00       1/1    main [1]
      0.01    0.00       1/1    fir(float*, float*, float*, int, int) [2]
      0.00    0.00       1/1    fir_opt(float*, float*, float*, int, int) [3]
      0.00    0.00       1/1    designLPF(float*, int, float, float) [14]
      0.00    0.00       1/1    fir_neon(float*, float*, float*, int, int) [13]
-----
[2]    66.7    0.02    0.00       1/1    main [1]
      0.02    0.00       1    fir(float*, float*, float*, int, int) [2]
-----
[3]    33.3    0.01    0.00       1/1    main [1]
      0.01    0.00       1    fir_opt(float*, float*, float*, int, int) [3]
-----
[11]   0.0     0.00    0.00       1/1    __libc_csu_init [20]
      0.00    0.00       1    _GLOBAL__sub_I_Z3firPfs_S_ii [11]
-----
[12]   0.0     0.00    0.00       1/1    __libc_csu_init [20]
      0.00    0.00       1    _GLOBAL__sub_I_main [12]
-----
[13]   0.0     0.00    0.00       1/1    main [1]
      0.00    0.00       1    fir_neon(float*, float*, float*, int, int) [13]
-----
[14]   0.0     0.00    0.00       1/1    main [1]
      0.00    0.00       1    designLPF(float*, int, float, float) [14]
-----

Index by function name
[11] _GLOBAL__sub_I_Z3firPfs_S_ii [2] fir(float*, float*, float*, int, int) [13] fir_neon(float*, float*, float*, int, int)
[12] _GLOBAL__sub_I_main [3] fir_opt(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
wes-237b@ubuntu:~/lab/lab3_pynq$

```

Compilation option O3

```

[wes-237b@ubuntu:~/lab/lab3_pynq$ sudo gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
%   cumulative   self           self       total
time  seconds    seconds   calls   ms/call  ms/call  name
66.89    0.02    0.02         1     20.07    20.07  fir(float*, float*, float*, int, int)
33.45    0.03    0.01         1     10.03    10.03  fir_opt(float*, float*, float*, int, int)
0.00    0.03    0.00         1      0.00     0.00  _GLOBAL__sub_I_Z3firPfS_S_ii
0.00    0.03    0.00         1      0.00     0.00  _GLOBAL__sub_I_main
0.00    0.03    0.00         1      0.00     0.00  fir_neon(float*, float*, float*, int, int)
0.00    0.03    0.00         1      0.00     0.00  designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 2 byte(s) for 33.22% of 0.03 seconds

index % time    self  children    called    name
-----
[1]  100.0    0.00    0.03        1/1    <spontaneous>
      0.02    0.00        1/1    main [1]
      0.01    0.00        1/1    fir(float*, float*, float*, int, int) [2]
      0.00    0.00        1/1    fir_opt(float*, float*, float*, int, int) [3]
      0.00    0.00        1/1    designLPF(float*, int, float, float) [14]
      0.00    0.00        1/1    fir_neon(float*, float*, float*, int, int) [13]
-----
[2]   66.7    0.02    0.00        1/1    main [1]
      0.02    0.00        1     fir(float*, float*, float*, int, int) [2]
-----
[3]   33.3    0.01    0.00        1/1    main [1]
      0.01    0.00        1     fir_opt(float*, float*, float*, int, int) [3]
-----
[11]  0.0     0.00    0.00        1/1    __libc_csu_init [20]
      0.00    0.00        1     _GLOBAL__sub_I_Z3firPfS_S_ii [11]
-----
[12]  0.0     0.00    0.00        1/1    __libc_csu_init [20]
      0.00    0.00        1     _GLOBAL__sub_I_main [12]
-----
[13]  0.0     0.00    0.00        1/1    main [1]
      0.00    0.00        1     fir_neon(float*, float*, float*, int, int) [13]
-----
[14]  0.0     0.00    0.00        1/1    main [1]
      0.00    0.00        1     designLPF(float*, int, float, float) [14]
-----

Index by function name

[11] _GLOBAL__sub_I_Z3firPfS_S_ii [2] fir(float*, float*, float*, int, int) [13] fir_neon(float*, float*, float*, int, int)
[12] _GLOBAL__sub_I_main [3] fir_opt(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
[wes-237b@ubuntu:~/lab/lab3_pynq$ ]

```

Compilation option -Ofast

```
wes-237b@ubuntu:~/lab/lab3_pynq$ sudo gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
%   cumulative   self           self         total
time  seconds    seconds   calls   ms/call  ms/call  name
66.89    0.02    0.02        1    20.07    20.07  fir(float*, float*, float*, int, int)
33.45    0.03    0.01        1    10.03    10.03  fir_opt(float*, float*, float*, int, int)
0.00    0.03    0.00        1     0.00     0.00  _GLOBAL__sub_I_Z3firPfs_S_ii
0.00    0.03    0.00        1     0.00     0.00  _GLOBAL__sub_I_main
0.00    0.03    0.00        1     0.00     0.00  fir_neon(float*, float*, float*, int, int)
0.00    0.03    0.00        1     0.00     0.00  designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 2 byte(s) for 33.22% of 0.03 seconds

index % time    self  children    called    name
[1]   100.0    0.00    0.03        1/1    <spontaneous>
      0.02    0.00        1/1    main [1]
      0.01    0.00        1/1    fir(float*, float*, float*, int, int) [2]
      0.00    0.00        1/1    fir_opt(float*, float*, float*, int, int) [3]
      0.00    0.00        1/1    designLPF(float*, int, float, float) [14]
      0.00    0.00        1/1    fir_neon(float*, float*, float*, int, int) [13]
-----
[2]    66.7    0.02    0.00        1/1    main [1]
      0.02    0.00        1    fir(float*, float*, float*, int, int) [2]
-----
[3]    33.3    0.01    0.00        1/1    main [1]
      0.01    0.00        1    fir_opt(float*, float*, float*, int, int) [3]
-----
[11]   0.0     0.00    0.00        1/1    __libc_csu_init [20]
      0.00    0.00        1    _GLOBAL__sub_I_Z3firPfs_S_ii [11]
-----
[12]   0.0     0.00    0.00        1/1    __libc_csu_init [20]
      0.00    0.00        1    _GLOBAL__sub_I_main [12]
-----
[13]   0.0     0.00    0.00        1/1    main [1]
      0.00    0.00        1    fir_neon(float*, float*, float*, int, int) [13]
-----
[14]   0.0     0.00    0.00        1/1    main [1]
      0.00    0.00        1    designLPF(float*, int, float, float) [14]
-----

Index by function name
[11] _GLOBAL__sub_I_Z3firPfs_S_ii [2] fir(float*, float*, float*, int, int) [13] fir_neon(float*, float*, float*, int, int)
[12] _GLOBAL__sub_I_main [3] fir_opt(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
wes-237b@ubuntu:~/lab/lab3_pynq$
```

Table showing performance of all three functions based on compiler optimization level.

Function	O0		O1		O2		O3		Ofast	
	Time	% Execution	Time	% Execution	Time	% Execution	Time	% Execution	Time	% Execution
fir	0.02	50	0.02	57.1	0.02	66.7	0.02	66.7	0.02	66.7
fir_opt	0.01	25	0.01	14.3	0.01	33.3	0.01	33.5	0.01	33.3
fir_neon	0.01	25	0.01	28.6	0.0	0.0	0.0	0.0	0.00	0.0

Part 1:2

Optimization: O0

Perf command: stat -d, along with time command. While time command gave total execution time, the perf didn't give meaning full information either. I am not getting demangled function name.

```
make: *** [clean] Error 1
[wes-237b@ubuntu:~/lab/lab3_pynq$ make
g++ -O0 -std=c++11 -O1 -ftree-vectorize -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
[wes-237b@ubuntu:~/lab/lab3_pynq$ sudo perf stat -d ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

Performance counter stats for './lab3_fir':

      174.214144 task-clock (msec)    #    0.994 CPUs utilized
           7    context-switches      #    0.040 K/sec
           0    cpu-migrations        #    0.000 K/sec
          597    page-faults          #    0.003 M/sec
    340,122,741 cycles                 #    1.952 GHz
    520,582,627 instructions          #    1.53 insn per cycle
<not supported> branches
    1,088,047 branch-misses
    180,957,992 L1-dcache-loads        # 1038.710 M/sec
    1,094,451 L1-dcache-load-misses   #   0.60% of all L1-dcache hits
<not supported> LLC-loads
<not supported> LLC-load-misses

      0.175339504 seconds time elapsed

[wes-237b@ubuntu:~/lab/lab3_pynq$ time ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

real    0m0.181s
user    0m0.168s
sys     0m0.008s
[wes-237b@ubuntu:~/lab/lab3_pynq$ ]
```

Optimization: O1

```
g++ -O1 -std=c++11 -O1 -ftree-vectorize -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
wes-237b@ubuntu:~/lab/lab3_pynq$ sudo perf stat -d ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

Performance counter stats for './lab3_fir':

    170.425952      task-clock (msec)          #    0.990 CPUs utilized
         19        context-switches          #    0.111 K/sec
          0        cpu-migrations            #    0.000 K/sec
        597        page-faults              #    0.004 M/sec
  338,402,875      cycles                   #    1.986 GHz
  520,459,047      instructions             #    1.54  insn per cycle
<not supported>   branches
   1,073,325       branch-misses
  180,946,657      L1-dcache-loads           # 1061.732 M/sec
   231,237         L1-dcache-load-misses     #    0.13% of all L1-dcache hits
<not supported>   LLC-loads
<not supported>   LLC-load-misses

    0.172091079 seconds time elapsed

wes-237b@ubuntu:~/lab/lab3_pynq$ time ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

real    0m0.183s
user    0m0.168s
sys     0m0.012s
wes-237b@ubuntu:~/lab/lab3_pynq$
```

Optimization: 02

```
[wes-237b@ubuntu:~/lab/lab3_pynq$ make
g++ -O2 -std=c++11 -O1 -ftree-vectorize -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
[wes-237b@ubuntu:~/lab/lab3_pynq$ sudo perf stat -d ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

Performance counter stats for './lab3_fir':

      167.041440      task-clock (msec)          #    0.995 CPUs utilized
           1         context-switches          #    0.006 K/sec
           0         cpu-migrations            #    0.000 K/sec
          595         page-faults              #    0.004 M/sec
    337,756,293       cycles                    #    2.022 GHz
    520,398,225       instructions              #    1.54   insn per cycle
<not supported>     branches
    1,030,412         branch-misses
    180,890,182       L1-dcache-loads           # 1082.906 M/sec
      852,302        L1-dcache-load-misses     #    0.47% of all L1-dcache hits
<not supported>     LLC-loads
<not supported>     LLC-load-misses

      0.167917642 seconds time elapsed

[wes-237b@ubuntu:~/lab/lab3_pynq$ time ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

real    0m0.177s
user    0m0.164s
sys     0m0.012s
[wes-237b@ubuntu:~/lab/lab3_pynq$
```


Optimization: O3

```
[wes-237b@ubuntu:~/lab/lab3_pynq$ make
g++ -O3 -std=c++11 -O1 -ftree-vectorize -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
[wes-237b@ubuntu:~/lab/lab3_pynq$ sudo perf stat -d ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

Performance counter stats for './lab3_fir':

      167.472032      task-clock (msec)          #    0.995 CPUs utilized
           0          context-switches          #    0.000 K/sec
           0          cpu-migrations            #    0.000 K/sec
          598         page-faults               #    0.004 M/sec
    334,086,024        cycles                    #    1.995 GHz
    520,420,653        instructions              #    1.56   insn per cycle
<not supported>      branches
    1,032,388          branch-misses
    180,905,867        L1-dcache-loads           # 1080.215 M/sec
    1,286,610          L1-dcache-load-misses     #    0.71% of all L1-dcache hits
<not supported>      LLC-loads
<not supported>      LLC-load-misses

      0.168298629 seconds time elapsed

[wes-237b@ubuntu:~/lab/lab3_pynq$ time ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

real    0m0.180s
user    0m0.168s
sys     0m0.012s
```

Optimization: Ofast

```

[wes-237b@ubuntu:~/lab/lab3_pynq$ make
g++ -Ofast -std=c++11 -O1 -ftree-vectorize -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -
[wes-237b@ubuntu:~/lab/lab3_pynq$ sudo perf stat -d ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

Performance counter stats for './lab3_fir':

      169.386016      task-clock (msec)      #    0.995 CPUs utilized
           0          context-switches      #    0.000 K/sec
           0          cpu-migrations       #    0.000 K/sec
          597         page-faults          #    0.004 M/sec
    338,748,231       cycles                #    2.000 GHz
    520,431,835       instructions          #    1.54   insn per cycle
<not supported>     branches
    1,031,771         branch-misses
    180,798,858       L1-dcache-loads       # 1067.378 M/sec
      463,707        L1-dcache-load-misses #    0.26% of all L1-dcache hits
<not supported>     LLC-loads
<not supported>     LLC-load-misses

      0.170195613 seconds time elapsed

[wes-237b@ubuntu:~/lab/lab3_pynq$ time ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00

real    0m0.182s
user    0m0.176s
sys     0m0.004s

```

Profiling works by changing how every function in the program is compiled so that when it is called, it will stash away some information about where it was called from. From this, the profiler can figure out what function called it, and can count how many times it was called. This change is made by the compiler when the program is compiled with the `-pg` option, which causes every function to call `mcount` (or `_mcount`, or `__mcount`, depending on the OS and compiler) as one of its first operations.

The `mcount` routine, included in the profiling library, is responsible for recording in an in-memory call graph table both its parent routine (the child) and its parent's parent. This is typically done by examining the stack frame to find both the address of the child, and the return address in the original parent. Since this is a very machine-dependant operation, `mcount` itself is typically a short assembly-language stub routine that extracts the required information, and then calls `__mcount_internal` (a normal C function) with two arguments - `frompc` and `selfpc`. `__mcount_internal` is responsible for maintaining the in-memory call graph, which records `frompc`, `selfpc`, and the number of times each of these call arcs was transversed. Because there are overhead, profiling does affect the program execution.

1:3 Profiling on Pynq

Optimization 00

```

[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ make
g++ -O0 -std=c++11 -mfpv=neon -O1 -ftree-vectorize -pg -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 %   cumulative   self           self      total
time  seconds    seconds   calls   ms/call  ms/call  name
42.86    0.06    0.06         1     60.00    60.00  fir(float*, float*, float*, int, int)
28.57    0.10    0.04         1     40.00    40.00  fir_opt(float*, float*, float*, int, int)
21.43    0.13    0.03         1     30.00    30.00  fir_neon(float*, float*, float*, int, int)
 7.14    0.14    0.01         1      0.00      0.00  main
 0.00    0.14    0.00         1      0.00      0.00  _GLOBAL__sub_I_Z3firPfS_S_ii
 0.00    0.14    0.00         1      0.00      0.00  _GLOBAL__sub_I_main
 0.00    0.14    0.00         1      0.00      0.00  designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 4 byte(s) for 7.14% of 0.14 seconds

index % time    self  children    called    name
[1]  100.0    0.01   0.13         1/1    <spontaneous>
      0.06   0.00         1/1    main [1]
      0.04   0.00         1/1    fir(float*, float*, float*, int, int) [2]
      0.03   0.00         1/1    fir_opt(float*, float*, float*, int, int) [3]
      0.00   0.00         1/1    fir_neon(float*, float*, float*, int, int) [4]
      0.00   0.00         1/1    designLPF(float*, int, float, float) [14]
-----
[2]   42.9    0.06   0.00         1/1    main [1]
      0.06   0.00         1    fir(float*, float*, float*, int, int) [2]
-----
[3]   28.6    0.04   0.00         1/1    main [1]
      0.04   0.00         1    fir_opt(float*, float*, float*, int, int) [3]
-----
[4]   21.4    0.03   0.00         1/1    main [1]
      0.03   0.00         1    fir_neon(float*, float*, float*, int, int) [4]
-----
[12]  0.0      0.00   0.00         1/1    __libc_csu_init [19]
      0.00   0.00         1    _GLOBAL__sub_I_Z3firPfS_S_ii [12]
-----
[13]  0.0      0.00   0.00         1/1    __libc_csu_init [19]
      0.00   0.00         1    _GLOBAL__sub_I_main [13]
-----
[14]  0.0      0.00   0.00         1/1    main [1]
      0.00   0.00         1    designLPF(float*, int, float, float) [14]
-----

Index by function name
[12] _GLOBAL__sub_I_Z3firPfS_S_ii [3] fir_opt(float*, float*, float*, int, int) [1] main
[13] _GLOBAL__sub_I_main [4] fir_neon(float*, float*, float*, int, int)
[2] fir(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$

```

Optimization O1

```

[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ make
g++ -O1 -std=c++11 -mfpu=neon -O1 -ftree-vectorize -pg -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ g^C log --online --graph
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 % cumulative self      self      total
time  seconds seconds  calls  ms/call  ms/call  name
33.33    0.05    0.05        1    50.00    50.00  fir(float*, float*, float*, int, int)
26.67    0.09    0.04        1    40.00    40.00  fir_opt(float*, float*, float*, int, int)
20.00    0.12    0.03        1    30.00    30.00  fir_neon(float*, float*, float*, int, int)
20.00    0.15    0.03        1    0.00    0.00  main
0.00    0.15    0.00        1    0.00    0.00  _GLOBAL__sub_I_Z3firPfS_S_ii
0.00    0.15    0.00        1    0.00    0.00  _GLOBAL__sub_I_main
0.00    0.15    0.00        1    0.00    0.00  designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 4 byte(s) for 6.67% of 0.15 seconds

index % time  self children  called  name
[1]  100.0    0.03  0.12        1  <spontaneous>
main [1]
    fir(float*, float*, float*, int, int) [2]
    fir_opt(float*, float*, float*, int, int) [3]
    fir_neon(float*, float*, float*, int, int) [4]
    designLPF(float*, int, float, float) [14]
-----
[2]    33.3    0.05  0.00        1  main [1]
    fir(float*, float*, float*, int, int) [2]
-----
[3]    26.7    0.04  0.00        1  main [1]
    fir_opt(float*, float*, float*, int, int) [3]
-----
[4]    20.0    0.03  0.00        1  main [1]
    fir_neon(float*, float*, float*, int, int) [4]
-----
[12]   0.0    0.00  0.00        1  __libc_csu_init [19]
    _GLOBAL__sub_I_Z3firPfS_S_ii [12]
-----
[13]   0.0    0.00  0.00        1  __libc_csu_init [19]
    _GLOBAL__sub_I_main [13]
-----
[14]   0.0    0.00  0.00        1  main [1]
    designLPF(float*, int, float, float) [14]
-----

Index by function name
[12] _GLOBAL__sub_I_Z3firPfS_S_ii [3] fir_opt(float*, float*, float*, int, int) [1] main
[13] _GLOBAL__sub_I_main [4] fir_neon(float*, float*, float*, int, int)
[2] fir(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)

```

Optimization O2

```
xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ make
g++ -O2 -std=c++11 -mfpu=neon -O1 -ftree-vectorize -pg -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00
xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ gprof -b lab3_fir gmon.out
Flat profile:
```

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self ms/call	total ms/call	name
40.00	0.06	0.06	1	60.00	60.00	fir(float*, float*, float*, int, int)
26.67	0.10	0.04	1	40.00	40.00	fir_opt(float*, float*, float*, int, int)
20.00	0.13	0.03	1	30.00	30.00	fir_neon(float*, float*, float*, int, int)
13.33	0.15	0.02				main
0.00	0.15	0.00	1	0.00	0.00	_GLOBAL__sub_I_Z3firPfS_S_ii
0.00	0.15	0.00	1	0.00	0.00	_GLOBAL__sub_I_main
0.00	0.15	0.00	1	0.00	0.00	designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 4 byte(s) for 6.67% of 0.15 seconds

index	% time	self	children	called	name
[1]	100.0	0.02	0.13		<spontaneous>
		0.06	0.00	1/1	main [1]
		0.04	0.00	1/1	fir(float*, float*, float*, int, int) [2]
		0.03	0.00	1/1	fir_opt(float*, float*, float*, int, int) [3]
		0.00	0.00	1/1	fir_neon(float*, float*, float*, int, int) [4]
		0.00	0.00	1/1	designLPF(float*, int, float, float) [14]
[2]	40.0	0.06	0.00	1/1	main [1]
		0.06	0.00	1	fir(float*, float*, float*, int, int) [2]
[3]	26.7	0.04	0.00	1/1	main [1]
		0.04	0.00	1	fir_opt(float*, float*, float*, int, int) [3]
[4]	20.0	0.03	0.00	1/1	main [1]
		0.03	0.00	1	fir_neon(float*, float*, float*, int, int) [4]
[12]	0.0	0.00	0.00	1/1	__libc_csu_init [19]
		0.00	0.00	1	_GLOBAL__sub_I_Z3firPfS_S_ii [12]
[13]	0.0	0.00	0.00	1/1	__libc_csu_init [19]
		0.00	0.00	1	_GLOBAL__sub_I_main [13]
[14]	0.0	0.00	0.00	1/1	main [1]
		0.00	0.00	1	designLPF(float*, int, float, float) [14]

Index by function name

```
[12] _GLOBAL__sub_I_Z3firPfS_S_ii [3] fir_opt(float*, float*, float*, int, int) [1] main
[13] _GLOBAL__sub_I_main [4] fir_neon(float*, float*, float*, int, int)
[2] fir(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
```

```
xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$
```

Optimization O3

```

[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ make
g++ -O3 -std=c++11 -mfp=neon -O1 -ftree-vectorize -pg -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
%   cumulative   self           self         total
time  seconds    seconds   calls   ms/call  ms/call  name
38.46    0.05     0.05        1     50.00    50.00  fir(float*, float*, float*, int, int)
30.77    0.09     0.04        1     40.00    40.00  fir_opt(float*, float*, float*, int, int)
30.77    0.13     0.04        1     40.00    40.00  fir_neon(float*, float*, float*, int, int)
 0.00    0.13     0.00        1      0.00     0.00  _GLOBAL__sub_I_Z3firPfS_S_ii
 0.00    0.13     0.00        1      0.00     0.00  _GLOBAL__sub_I_main
 0.00    0.13     0.00        1      0.00     0.00  designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 4 byte(s) for 7.69% of 0.13 seconds

index % time    self  children   called    name
[1]  100.0    0.00    0.13        1/1    <spontaneous>
      0.05    0.00        1/1    main [1]
      0.04    0.00        1/1    fir(float*, float*, float*, int, int) [2]
      0.04    0.00        1/1    fir_opt(float*, float*, float*, int, int) [3]
      0.00    0.00        1/1    fir_neon(float*, float*, float*, int, int) [4]
      0.00    0.00        1/1    designLPF(float*, int, float, float) [14]
-----
[2]   38.5    0.05    0.00        1/1    main [1]
      0.05    0.00        1    fir(float*, float*, float*, int, int) [2]
-----
[3]   30.8    0.04    0.00        1/1    main [1]
      0.04    0.00        1    fir_opt(float*, float*, float*, int, int) [3]
-----
[4]   30.8    0.04    0.00        1/1    main [1]
      0.04    0.00        1    fir_neon(float*, float*, float*, int, int) [4]
-----
[12]  0.0     0.00    0.00        1/1    __libc_csu_init [19]
      0.00    0.00        1    _GLOBAL__sub_I_Z3firPfS_S_ii [12]
-----
[13]  0.0     0.00    0.00        1/1    __libc_csu_init [19]
      0.00    0.00        1    _GLOBAL__sub_I_main [13]
-----
[14]  0.0     0.00    0.00        1/1    main [1]
      0.00    0.00        1    designLPF(float*, int, float, float) [14]
-----

Index by function name

[12] _GLOBAL__sub_I_Z3firPfS_S_ii [2] fir(float*, float*, float*, int, int) [4] fir_neon(float*, float*, float*, int, int)
[13] _GLOBAL__sub_I_main [3] fir_opt(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$

```

Optimization Ofast

```

[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ make
g++ -Ofast -std=c++11 -mfpu=neon -O1 -ftree-vectorize -pg -Iinclude src/fir.cpp src/main.cpp -o lab3_fir -lc
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ ./lab3_fir
RMSE_naive: 0.00
RMSE_opt: 0.00
RMSE_neon: 0.00
[xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$ gprof -b lab3_fir gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 % cumulative self      self      total
time  seconds seconds  calls  ms/call  ms/call  name
35.71    0.05    0.05        1    50.00    50.00  fir(float*, float*, float*, int, int)
28.57    0.09    0.04        1    40.00    40.00  fir_opt(float*, float*, float*, int, int)
21.43    0.12    0.03        1    30.00    30.00  fir_neon(float*, float*, float*, int, int)
14.29    0.14    0.02        1     0.00     0.00  main
 0.00    0.14    0.00        1     0.00     0.00  _GLOBAL__sub_I_Z3firPfS_S_ii
 0.00    0.14    0.00        1     0.00     0.00  _GLOBAL__sub_I_main
 0.00    0.14    0.00        1     0.00     0.00  designLPF(float*, int, float, float)

Call graph

granularity: each sample hit covers 4 byte(s) for 7.14% of 0.14 seconds

index % time    self  children   called    name
-----
[1]   100.0    0.02   0.12      1/1      <spontaneous>
      0.05   0.00      1/1      main [1]
      0.04   0.00      1/1      fir(float*, float*, float*, int, int) [2]
      0.03   0.00      1/1      fir_opt(float*, float*, float*, int, int) [3]
      0.00   0.00      1/1      fir_neon(float*, float*, float*, int, int) [4]
      0.00   0.00      1/1      designLPF(float*, int, float, float) [14]
-----
[2]    35.7    0.05   0.00      1/1      main [1]
      0.05   0.00      1      fir(float*, float*, float*, int, int) [2]
-----
[3]    28.6    0.04   0.00      1/1      main [1]
      0.04   0.00      1      fir_opt(float*, float*, float*, int, int) [3]
-----
[4]    21.4    0.03   0.00      1/1      main [1]
      0.03   0.00      1      fir_neon(float*, float*, float*, int, int) [4]
-----
[12]    0.0    0.00   0.00      1/1      __libc_csu_init [19]
      0.00   0.00      1      _GLOBAL__sub_I_Z3firPfS_S_ii [12]
-----
[13]    0.0    0.00   0.00      1/1      __libc_csu_init [19]
      0.00   0.00      1      _GLOBAL__sub_I_main [13]
-----
[14]    0.0    0.00   0.00      1/1      main [1]
      0.00   0.00      1      designLPF(float*, int, float, float) [14]
-----

Index by function name

[12] _GLOBAL__sub_I_Z3firPfS_S_ii [3] fir_opt(float*, float*, float*, int, int) [1] main
[13] _GLOBAL__sub_I_main [4] fir_neon(float*, float*, float*, int, int)
[2] fir(float*, float*, float*, int, int) [14] designLPF(float*, int, float, float)
xilinx@pynq:~/jupyter_notebooks/wes237b_labs/master/WES237B/Assignment3/lab3_pynq$

```

Yes, the platform matters. Jetson is faster than Pynq to execute the same program. Also because of the overhead of profiling, faster CPU would give better result.

Table showing performance of Pynq board

Function	O0		O1		O2		O3		Ofast	
	Time	% Execution	Time	% Execution	Time	% Execution	Time	% Execution	Time	% Execution
fir	0.06	42.9	0.05	33.33	0.06	40.00	0.05	38.5	0.05	35.7
fir_opt	0.04	28.6	0.09	26.57	0.10	26.67	0.04	30.8	0.04	28.6
fir_neon	0.03	21.4	0.12	20.00	0.13	20.00	0.04	30.8	0.03	21.4

Part 2

1. I used two approaches. One is using float32x4 vectors. Other is using int16x8 registers.
 - a. Float 32 approach: I loaded 4 bytes of input data to registers in 4 registers at once, and 4 bytes of kernels on other registers. I took a loop to multiply and accumulate based on block size. Block size is 4. So 8 operations are done through in loop. Final operation is done through normal c++ programming. I accessed required input data using array index operations and looping to fill the data.
 - b. Int16 approach. Here, 8 operations takes place simultaneously. 8 bytes are loaded on 8 registers. 8 bytes of kernels are loaded on other 8 registers. Multiply and accumulate the result. Because kernel size is 9. The remaining 1 operation is done normally. I accessed required input data using array index operations and looping to fill the data.
2. Board taken: **Jetson. Optimization level O2.**
 WIDTH = HEIGHT = 3072.
 Even with image size of 3072, OpenCV:Sobel gave output as 0.0 in GProf. I added timer.h done in lab4 and used it to measure time using linux timer.

Average time of Open CV: 0.02

2.1 Naïve Sobel filter:

```
[wes-237b@ubuntu:~/lab/sobel$ ls
display_output.ipynb hw3 image_outputs include input.raw Makefile objs src Untitled.ipynb view_images.ipynb
[wes-237b@ubuntu:~/lab/sobel$ ./hw3 1 3072
WES237B hw3

[wes-237b@ubuntu:~/lab/sobel$ sudo gprof -b hw3 gmon.out
[sudo] password for wes-237b:
Flat profile:

Each sample counts as 0.01 seconds.
   %   cumulative   self           total
time  seconds  seconds    calls   Ts/call   Ts/call  name
-----
100.00      0.34      0.34         1      0.00      0.00  sobel(cv::Mat const&, cv::Mat&)
0.00        0.34      0.00         1      0.00      0.00  _GLOBAL__sub_I__Z5sobelRKN2cv3MatERS0_
0.00        0.34      0.00         1      0.00      0.00  _GLOBAL__sub_I__Z5usagev
0.00        0.34      0.00         1      0.00      0.00  _init

Call graph

granularity: each sample hit covers 2 byte(s) for 2.94% of 0.34 seconds

index % time    self  children    called    name
-----
[1]   100.0    0.34    0.00        1/1    <spontaneous>
sobel(cv::Mat const&, cv::Mat&) [1]
-----
[10]    0.0    0.00    0.00        1/1    __libc_csu_init [43]
_GLOBAL__sub_I__Z5sobelRKN2cv3MatERS0_ [10]
-----
[11]    0.0    0.00    0.00        1/1    __libc_csu_init [43]
_GLOBAL__sub_I__Z5usagev [11]
-----
[12]    0.0    0.00    0.00        1/1    __libc_csu_init [43]
_init [12]
-----

Index by function name

[10] _GLOBAL__sub_I__Z5sobelRKN2cv3MatERS0_ [1] sobel(cv::Mat const&, cv::Mat&)
[11] _GLOBAL__sub_I__Z5usagev [12] _init
[wes-237b@ubuntu:~/lab/sobel$ ls
display_output.ipynb gmon.out hw3 image_outputs include input.raw Makefile objs src Untitled.ipynb view_images.ipynb
[wes-237b@ubuntu:~/lab/sobel$ rm gmon.out
[wes-237b@ubuntu:~/lab/sobel$ ls
display_output.ipynb hw3 image_outputs include input.raw Makefile objs src Untitled.ipynb view_images.ipynb
[wes-237b@ubuntu:~/lab/sobel$ ./hw3 1 3072
WES237B hw3
```

2.2 Optimized Sobel filter: Loop unrolled

Note: I have implemented two ways of loop unrolled. In first way under the macro LOOP_UNROLL_1, because kernel size is 3*3. I iterate over kernel from 0-3 and do 3 operations simultaneously.

In other approach, I do not iterate over kernel. Because there are 9 operations in total. I perform all nine operations under two loops, iterating over input image by accessing required indices. This approach gives better result.

LOOP_UNROLL_1 output

```
wes-237b@ubuntu:~/lab/sobel$ make
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/mat.cpp -o objs/mat.o
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/main.cpp -o objs/main.o
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/student_sobel.cpp -o objs/student_sobel.o
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/main.cpp -o objs/main.o
3 -lc -lopencv_core -lopencv_highgui -lopencv_imgproc -lopencv_imgcodecs -lopencv_videoio
wes-237b@ubuntu:~/lab/sobel$ ./hw3 2 3072
WES237B hw3

wes-237b@ubuntu:~/lab/sobel$ sudo gprof -b hw3 gmon.out
[sudo] password for wes-237b:
Flat profile:

Each sample counts as 0.01 seconds.
%   cumulative   self           calls     self   total    name
time  seconds    seconds             Ts/call   Ts/call             name
100.00      0.27      0.27              1      0.00    0.00  sobel_unroll(cv::Mat const&, cv::Mat&)
0.00      0.27      0.00              1      0.00    0.00  _GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_
0.00      0.27      0.00              1      0.00    0.00  _GLOBAL__sub_I_Z5usagev
0.00      0.27      0.00              1      0.00    0.00  _init

Call graph

granularity: each sample hit covers 2 byte(s) for 3.70% of 0.27 seconds

index % time    self  children    called    name
-----
[1]   100.0     0.27    0.00          1    <spontaneous>
sobel_unroll(cv::Mat const&, cv::Mat&) [1]
-----
[10]    0.0     0.00    0.00          1    __libc_csu_init [43]
_GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_ [10]
-----
[11]    0.0     0.00    0.00          1    __libc_csu_init [43]
_GLOBAL__sub_I_Z5usagev [11]
-----
[12]    0.0     0.00    0.00          1    __libc_csu_init [43]
_init [12]
-----

Index by function name

[10] _GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_ [1] sobel_unroll(cv::Mat const&, cv::Mat&)
[11] _GLOBAL__sub_I_Z5usagev [12] _init
wes-237b@ubuntu:~/lab/sobel$
```

LOOP_UNROLL_1 macro disabled output

```

[wes-237b@ubuntu:~/lab/sobel$ make
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/mat.cpp -o objs/mat.o
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/main.cpp -o objs/main.o
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/student_sobel.cpp -o objs/student_sobel.o
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/mat.o objs/main.o objs/student_sobel.o -o hw3
3 -lc -lopencv_core -lopencv_highgui -lopencv_imgproc -lopencv_imgcodecs -lopencv_videoio
[wes-237b@ubuntu:~/lab/sobel$ ./hw3 2 3072
WES237B hw3

[wes-237b@ubuntu:~/lab/sobel$ sudo gprof -b hw3 gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
   %   cumulative   self   total     name
time  seconds  seconds  calls  Ts/call  Ts/call  name
100.00    0.16    0.16         1    0.00    0.00  sobel_unroll(cv::Mat const&, cv::Mat&)
  0.00    0.16    0.00         1    0.00    0.00  _GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_
  0.00    0.16    0.00         1    0.00    0.00  _GLOBAL__sub_I_Z5usagev
  0.00    0.16    0.00         1    0.00    0.00  _init

Call graph

granularity: each sample hit covers 4 byte(s) for 6.25% of 0.16 seconds

index % time    self  children   called    name
-----
[1]   100.0    0.16    0.00      1/1    <spontaneous>
      0.00    0.00      1/1    sobel_unroll(cv::Mat const&, cv::Mat&) [1]
[10]    0.0    0.00    0.00      1/1    __libc_csu_init [43]
      0.00    0.00      1/1    _GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_ [10]
[11]    0.0    0.00    0.00      1/1    __libc_csu_init [43]
      0.00    0.00      1/1    _GLOBAL__sub_I_Z5usagev [11]
[12]    0.0    0.00    0.00      1/1    __libc_csu_init [43]
      0.00    0.00      1/1    _init [12]

Index by function name

[10] _GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_ [1] sobel_unroll(cv::Mat const&, cv::Mat&)
[11] _GLOBAL__sub_I_Z5usagev [12] _init
[wes-237b@ubuntu:~/lab/sobel$

```

2.3 Neon method: I implemented using 4 vector operations at one go and 8 vector operations in one go

a) FLOAT_IMPL: 4 vector operations in one go.

```

[11] _GLOBAL__sub_I_Z5usagev [12] _init
[wes-237b@ubuntu:~/lab/sobel$ ls
display_output.ipynb  gmon.out  hw3  image_outputs  include  input.raw  Makefile  objs  src  Untitled.ipynb  view_images.ipynb
[wes-237b@ubuntu:~/lab/sobel$
[wes-237b@ubuntu:~/lab/sobel$ ./hw3 3 3072
WES237B hw3

[wes-237b@ubuntu:~/lab/sobel$ sudo gprof -b hw3 gmon.out
[sudo] password for wes-237b:
Flat profile:

Each sample counts as 0.01 seconds.
%   cumulative   self           self         total
time  seconds    seconds    calls   Ts/call   Ts/call   name
100.00    0.58    0.58                1      0.00    0.00  sobel_neon(cv::Mat const&, cv::Mat&)
  0.00    0.58    0.00                1      0.00    0.00  _GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_
  0.00    0.58    0.00                1      0.00    0.00  _GLOBAL__sub_I_Z5usagev
  0.00    0.58    0.00                1      0.00    0.00  _init

Call graph

granularity: each sample hit covers 4 byte(s) for 1.72% of 0.58 seconds

index % time    self  children    called    name
-----
[1]   100.0    0.58    0.00        1/1    <spontaneous>
      0.00    0.00        1/1    sobel_neon(cv::Mat const&, cv::Mat&) [1]
[10]   0.0     0.00    0.00        1/1    __libc_csu_init [43]
      0.00    0.00        1/1    _GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_ [10]
[11]   0.0     0.00    0.00        1/1    __libc_csu_init [43]
      0.00    0.00        1/1    _GLOBAL__sub_I_Z5usagev [11]
[12]   0.0     0.00    0.00        1/1    __libc_csu_init [43]
      0.00    0.00        1/1    _init [12]

Index by function name

[10] _GLOBAL__sub_I_Z5sobelRKN2cv3MatERS0_ [1] sobel_neon(cv::Mat const&, cv::Mat&)
[11] _GLOBAL__sub_I_Z5usagev [12] _init
[wes-237b@ubuntu:~/lab/sobel$

```

b) FLOAT_IMPL disabled: 8 vector operations in one loop.

```
[11] _GLOBAL__sub_I__Z5usagev [12] _init
[wes-237b@ubuntu:~/lab/sobel$ rm gmon.out
[wes-237b@ubuntu:~/lab/sobel$ make
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude -c src/student_sobel.cpp -o objs/student_sobel.o
g++ -O2 -ftree-vectorize -std=c++11 -I/usr/include/opencv4/ -pg -Iinclude objs/mat.o objs/main.o objs/student_sobel.o -o hw
3 -lc -lopencv_core -lopencv_highgui -lopencv_imgproc -lopencv_imgcodecs -lopencv_videoio
[wes-237b@ubuntu:~/lab/sobel$ ./hw3 3 3072
WES237B hw3

[wes-237b@ubuntu:~/lab/sobel$ sudo gprof -b hw3 gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 %   cumulative   self           total
time  seconds    seconds    calls   Ts/call   Ts/call  name
100.00    0.37    0.37              1     0.00     0.00  sobel_neon(cv::Mat const&, cv::Mat&)
 0.00    0.37    0.00              1     0.00     0.00  _GLOBAL__sub_I__Z5sobelRKN2cv3MatERS0_
 0.00    0.37    0.00              1     0.00     0.00  _GLOBAL__sub_I__Z5usagev
 0.00    0.37    0.00              1     0.00     0.00  _init

      Call graph

granularity: each sample hit covers 2 byte(s) for 2.70% of 0.37 seconds

index % time    self  children    called    name
-----
[1]   100.0    0.37    0.00          1/1  <spontaneous>
      sobel_neon(cv::Mat const&, cv::Mat&) [1]
-----
[10]    0.0    0.00    0.00          1/1  __libc_csu_init [43]
      _GLOBAL__sub_I__Z5sobelRKN2cv3MatERS0_ [10]
-----
[11]    0.0    0.00    0.00          1/1  __libc_csu_init [43]
      _GLOBAL__sub_I__Z5usagev [11]
-----
[12]    0.0    0.00    0.00          1/1  __libc_csu_init [43]
      _init [12]
-----

Index by function name

[10] _GLOBAL__sub_I__Z5sobelRKN2cv3MatERS0_ [1] sobel_neon(cv::Mat const&, cv::Mat&)
[11] _GLOBAL__sub_I__Z5usagev [12] _init
[wes-237b@ubuntu:~/lab/sobel$ ]
```

Approach with 8 operations in one go is fast.

2:3

Table showing performance on Jetson board with optimization option 2 Image size 3072

Function	O2		Alternate approach	
	Time	% Execution	Time	% Execution
Open CV	0.02	NA	NA	NA
Sobel_naive	0.34	100	NA	NA
Sobel_unrolled	0.27	100	0.16	100
Sobel Neon	0.58	100	0.37	100

Note Alternate approach led to better results. In sober unrolled, I am manually doing all nine operations together using proper indices access. In Neon approach, I am using 8 operations simultaneously.

Part 3:

1. What does the **global** flag mean?

`__global__` specifier means that the procedure runs on device (GPU)

2. Describe `<<<4,32>>>` in terms of threads blocks and grids.
`<<<...>>>` describes kernel configuration. Multiple threads are organized into thread block. Multiple thread blocks are organized into Grid.
Above configuration means, 4 thread blocks. Each thread block has 32 parallel threads.