ECE-GY-9413 Course Project

Part 2 - Submission due on March 8, 2024 at 11:59 PM

The second part of the course project requires you to implement multiple test programs in assembly to run on your vector simulator. Details follow below for the test programs to be implemented.

Fully Connected Layer:

The first test program is going to be a Fully Connected Layer. The fully connected layer should be 256x256. It is a 256 vector by 256x256 matrix multiplication.

$FCC = a * W + b,$

where,

a = {0, 1, 2, 3, ... 255}

```
W = { {-1, 1, -1, 1, . . . .  1},
      {1, -1, 1, -1, . . . . -1},
      :                         ;
      :                         ;
      {1, -1, 1, -1, . . . . -1}
    }
```

b = {128, 256, 128, ... 256}

Convolution Layer:

The second test program is going to be a Convolution Layer. The 2D convolution layer should have an input frame of size 256x256 and a kernel of size 3x3 with zero padding and a stride of 2.

$CONV = (f * g)_t$

```
f = { {-1, 1, -1, 1, . . . .  1},
      {1, -1, 1, -1, . . . . -1},
      :                         ;
      :                         ;
      {1, -1, 1, -1, . . . . -1}
    }
g = { {  3,  2,  1},
      { -2,  3,  2},
      { -1, -2,  3}
    }
```

Fast Fourier Transform (FFT):

The third test program is going to be a FFT. You should implement a 128-point FFT. The provided file “twiddle_factors.txt” contains precomputed values of Twiddle factors from W^0_{128} to W^{63}_{128}. These precomputed values should be stored in the VDMEM. Since our simulator only deals with integers, the Twiddle factors will have to be scaled by a factor of 1000 (subsequently any products should be scaled down by 1000).

Input to the FFT is:

x = {1024+512j, 1024-512j, 512+1024j, 512-1024j, ... (32 times)}

Guidelines:

Input: Each test function should have the following input files:

- Code.asm: The file should contain your assembly code for the test functions.
    - **Add comments to your code. Comments should start with a hash sign (#) and can be inline. Modify your simulator accordingly so it can parse this.**
- SDMEM.txt: The file should contain the initial state of the SDMEM containing the data required for the test function in integer format. Each line in this file represents one word (32 bit) of data in the SDMEM.
- VDMEM.txt: The file should contain the initial state of the VDMEM containing the data required for the test function in integer format. Each line in this file represents one word (32 bit) of data in the VDMEM.

Output: Each test function should have the following output files:

- VRF.txt: The file should show the final state of the Vector Register File after the execution of all the instructions in the input Code.asm file. Each line should contain a comma separated list of integer values showing all the elements of a vector register.
- SRF.txt: The file should show the final state of the Scalar Register File after the execution of all the instructions in the input Code.asm file. Each line should contain an integer value showing a scalar register data.
- SDMEMOP.txt: The file should contain the final state of the SDMEM after the execution of the Code.asm file. The format should be the same as the input SDMEM.txt.
- VDMEMOP.txt: The file should contain the final state of the VDMEM after the execution of the Code.asm file. The format should be the same as the input VDMEM.txt.

Evaluation Criteria:

Your implementation will be evaluated using the test files submitted and the specified inputs above. The output from your implementation will be matched against a golden test result. Points will be given based on the following criteria:

- Correct Fully Connected Layer (commented): 34%
- Correct Convolution Layer (commented): 33%
- Correct FFT (commented): 33%

Deliverables:

1. Your Python script(s) implementing the functional simulator. The name of the main script should be <netid>_funcsimulator.py.
2. Three test folders: One for each test program with your Code.asm, input files and outputs.
3. Code comments; code without comments will not receive credit.

Please zip the files and use your net id in the final zip file name.