CS 259 HW 2

HLS Practice: Matrix Multiplication

Instructor: Prof. Jason Cong

January 25th, 2017

1 Description

Your assignment is to accelerate the matrix multiplication using Vivado HLS. Listing 1 shows the example code you can start with.

```
typedef float data_t;

void mat_mul(data_t* a, data_t* b, data_t* c, int M, int N, int K)
#pragma HLS INTERFACE m_axi port=a offset=slave bundle=gmem
#pragma HLS INTERFACE m_axi port=b offset=slave bundle=gmem
#pragma HLS INTERFACE m_axi port=c offset=slave bundle=gmem
#pragma HLS INTERFACE s_axilite port=a bundle=control
#pragma HLS INTERFACE s_axilite port=b bundle=control
#pragma HLS INTERFACE s_axilite port=c bundle=control
#pragma HLS INTERFACE s_axilite port=M bundle=control
#pragma HLS INTERFACE s_axilite port=N bundle=control
#pragma HLS INTERFACE s_axilite port=K bundle=control
#pragma HLS INTERFACE s_axilite port=return bundle=control

// initialize matrix C
for (int i = 0; i < M; i++) {
  for (int j = 0; j < N; j++) {
        c[i*N + j] = 0;
  }
}

// computation
for (int k = 0; k < K; k++) {
  for (int i = 0; i < M; i++) {
    for (int j = 0; j < N; j++) {
      c[i*N + j] += a[i*K + k] * b[k*N + j];
    }
  }
}
```

Listing 1: Example code of matrix multiplication.

In the example code, we calculate $C = A \times B$, where $A \in \mathbb{R}^{M \times K}$, $B \in \mathbb{R}^{K \times N}$, and $C \in \mathbb{R}^{M \times N}$. Suppose that we have already stored three matrices in the external DRAMs. For the top function *mat_mul*, we pass the pointers a, b, and c which point to the starting address of each matrix. All the matrices are stored in row-major.

In the example code, we have already added the pragmas for the kernel interface. For more details about AXI bus, please refer to Xilinx UG902.

We have allocated 20GB space for everyone on Seasnet servers (lnxsrv01-lnxsrv05). Please work under the directory: `/w/tempcs259/<your_loginname>` . To begin with, execute the following two commands to set up your environment. You can also add them to your `bashrc`.

```
$ export LM_LICENSE_FILE=/usr/local/cs133/Xilinx/SDAccel/Xilinx_lnxsrv<01-05>.lic
```

```
$ source /usr/local/cs133/Xilinx/SDAccel/2015.4_Yao/settings64.sh
```

Then, use command `$ vivado_hls &` or `$ vivado_hls mm.tcl` to start Vivado HLS.

2 Submission

You have to submit a brief report which summarizes:

- optimization strategies you applied in this project.
- acceleration results of your kernel, including the latency and II of your design ($M = N = K = 128$).

You will also need to submit your optimized code with your test bench file. Please verify the correctness of your kernel before submission.

Your final submission should be a tarball which contains the following files:

```
<Your UID>.tar.gz
├── mm.cpp
├── mm.h(if existed)
├── tb_mm.cpp
└── report
```

3 References

We strongly recommend you to read this paper before starting your project. This paper is not published yet, please do not distribute it outside the class.

Best-Effort FPGA Programming: A Few Steps Can Go a Long Way. http://cadlab.cs.ucla.edu/ cody/public/fpga-programmability.pdf