Report for Systolic Array Matrix Multiplication

Wan Haochuan

I. Introduction

This report introduce the total workflow of the weight stationary systolic array and detail implementation of three sub-modules, which are im2col, pe and systolic_array. The im2col module reads data from the memory, performs the im2col conversion and saves converted data to memory. The basic function of PE is calculating the dot products of the rows and columns, and streaming the inputs to its neighbors. The systolic array is constructed by instantiate PE modules. Systolic arrays are widely used in many hardware accelerator design and one of the well-recognized examples is the Google's tensor processing unit (TPU) [1].

II. Implementation Details

Each module use asynchronous reset.

A. *im2col*

The im2col module mainly has 6 ports. **data_rd** and **addr_rd** control the funtion of data reading from memory. **data_wr**, **addr_wr** and **mem_wr_en** control the function of data writing to memory. **done** represent the state of the im2col module.

The im2col has a state machine with 3 mainly states, which are READ state, WRITE state and DONE state. The READ state will use the read ports to get data from memory to the register array in the im2col. After the READ state, the state machine will go to the WRITE state, which has two for loops. The first for loop traverses the kernel window by each pixel in the image and the second loop traverses each point in the kernel window. In the WRITE state, the point will be judged weather it is in the image range. If it inside the image, data from the register array will be written to the memory, if not, zero will be written. After WRITE state, the state machine go to DONE state and pull up **done** signal.

B. *pe*

The pe module mainly has ports of **w**, **x_in**, **y_in**, **x_out** and **y_out**. At each clock, **x_out** get the data of **x_in** and **y_out** get the data of **y_in** + **w** × **x_in**.

C. *systolic_array*

The systolic_array has ports of **X**, **W**, **Y**, **valid** and **done**. The whole module is composed by pe. In each pe, the below pe get **y_in** from the upper pe, the right pe get **x_in** from the left pe and get **w** from **W** port directly. For pe modules in the first column, they get **x_in** from **X** port after passing thought buffers according to the row number. For pe modules in the last row, they output **y_out** to **Y** port after passing thought buffers according to the column number. When the first row of **Y** is ready, pull up the **valid** port and hold high. When the last row of **Y** is ready, pull up the **done** port.

III. Conclusions

In conclusion, a weight stationary systolic array with three main modules, which are im2col, pe and systolic_array, is implemented successfully. For calculation of $M \times N$ image with X $K \times K$ kernel, it will take $M \times N$ cycles to read data from memory, $M \times N \times K \times K$ cycles to write data to memory by im2col and $M \times N + K \times K + X$ cycles to finish computation by systolic_array. It takes $2 \times M \times N + M \times N \times K \times K + K \times K + X$ cycles totally and get correct result in every testcase.

References

[1] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon, "In-datacenter performance analysis of a tensor processing unit," in *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, 2017, pp. 1–12.