University of Tehran

Electrical and Computer Engineering Department

ECE (8101) 342

Object Oriented Modeling of Electronic Circuits – Spring 1401-02

Homework 4: SystemC Design and Modeling
Due Date: Ordibehesht 04

2D Weight-Stationary Systolic Architecture for Matrix Multiplication

Matrix multiplication is the fundamental building block of many algorithms such as data analytics and neural networks. The sequential implementation of this operation is very time-consuming for large matrices. The systolic array is proposed as a low-cost solution for matrix multiplication. Along this line, a 2D systolic array forms the heart of the Matrix Multiplier Unit (MXU) on the Google TPU and the new deep-learning FPGAs from Xilinx.

A systolic architecture consists of a set of interconnected cells, also called Processing Elements (PEs), each capable of performing a single Multiply-And-Accumulate (MAC) Computation. The general view of a systolic array can be seen in Figure 1. As shown, data flows directly between cells in a pipelined fashion, and communications with the outside world occur only at the boundary cells. Based on the way the operands of the matrix multiplication are being handled during execution, two categories of systolic arrays have been proposed: non-stationary and stationary. While in the non-stationary architecture, both operands of the matrix multiplication flow through the MAC units, in the stationary architecture, only one of them flows.

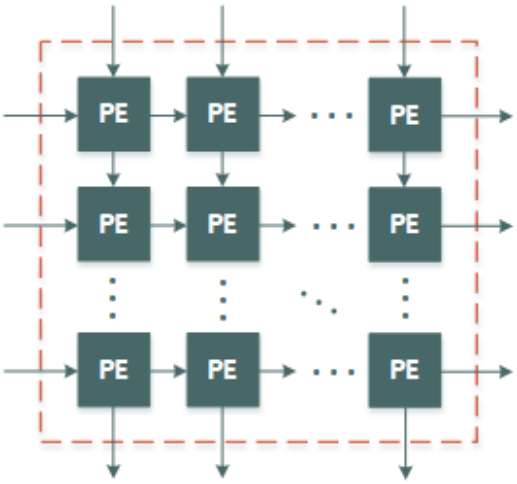

Figure 1 – An overall view of the systolic array

A more popular type of systolic array for matrix multiplication is the TPU-style Stationary Systolic Array (TSSA), which is the architecture of the systolic array in TPU. TSSA is also called weight stationary and has been implemented for neural networks. In the weight-stationary architecture, the PEs keep the weight inputs in their registers and pass through their outputs. Therefore, before starting the multiplications, the weight inputs must be loaded to the registers of each PE. In a nutshell, in the weight-stationary architecture, the weights stay stationary and the inputs are streamed in. The input data (*D*) is moved through the systolic array horizontally. Figure 2 shows the operand handling in the weight-stationary architecture.

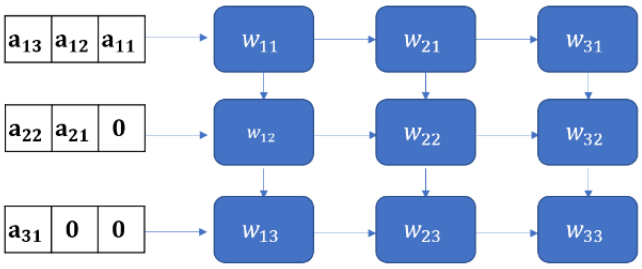

Figure 2 – Operands in weight-stationary systolic array

The systolic array circuit starts its operation with a positive pulse on the *start* signal. When the operation starts, the weight matrix is loaded in the PEs. After that, the PEs starts the calculation process by asserting a *busyPE* signal. The results can be collected from the bottom cells in several consecutive clock cycles. When the output matrix is completed, the *done* signal becomes 1. This signal remains active until the next time that the *start* signal becomes 1.

In the cases that the input matrices are large and cannot be mapped on the systolic array once, input matrices are partitioned and the systolic architecture is reused in several iterations. As a result, the iteration number must be checked before issuing the *done* signal.

A systolic cell (PE) consists of two 8-bit inputs named *Di* and *Wi*, a 24-bit input *Si*, and two control inputs *startPE* and *busyPE* for weights preloading and MAC computation. Inside the cell, there are two registers to latch the inputs *Di*, *Wi*, along with a multiplier to calculate the *Di*Wi* product, and an adder to add this product to the coming partial sum *Si*. There is also a result register that stores partial sum results, i.e., *So*. When a systolic cell sees the *startPE* signal goes positive, it loads the weight register. When the systolic cell detects the *busyPE* signal, it starts its MAC calculation.

Structural RTL Modeling Phase:

In this phase, you are to implement a structural RTL model of a configurable 2D weight-stationary systolic array. The size of the systolic array (the number of PEs) is determined through a template class.

A) Design and describe the RTL model of a PE in SystemC.

- Show the schematic diagram of the datapath and controller of the PE.
- Write the PE datapath and controller in SystemC.
- Write a testbench and test your PE using different scenarios.

B) Design and describe the RTL model of the systolic array by putting together PEs.

- Show the schematic diagram of the datapath and controller of the systolic array.
- Write the datapath and controller of the systolic array in SystemC.
- Write a testbench and test your PE using the following scenarios:
  - ✓ The size of input matrices is smaller than the systolic array size. Therefore, some PEs are not used.
  - ✓ The size of input matrices is equal to the systolic array size.
  - ✓ The size of input matrices is bigger than the systolic array size. Therefore, the systolic array is reused.

RTL Bus Functional Modeling Phase:

In this phase, you are to implement the BFM model of a configurable 2D weight-stationary systolic array. The size of the systolic array (the number of PEs) is determined through a template class.

A) Describe the BFM model of the systolic array in SystemC.

B) Write a testbench and test your PE using the following scenarios:

- ✓ The size of input matrices is smaller than the systolic array size. Therefore, some PEs are not used.
- ✓ The size of input matrices is equal to the systolic array size.
- ✓ The size of input matrices is bigger than the systolic array size. Therefore, the systolic array is reused.

C) Compare the waveforms of the structural and functional models.

Deliverables:

1. All SystemC codes with proper naming
2. A complete report containing
   - Schematic diagrams drawing in Visio or other visualization tools,
   - Enough design illustration and description,
   - Simulation results, input data, and output justification.
