ESE 461: Design Automation for Integrated Circuit Systems

Lab4: MAC

Due: Nov-15th, Submitted by Github

1. Overview

In this lab, you need to implement the dot product of two vectors **a** = [a1, a2, …, an] and **b** = [b1, b2, …, bn] by MAC unit. You need to implement 2 versions, one design without pipeline and one with pipeline, then compare the synthesized report on area, power and timing.

Input: clk, reset

Output: $dot\ product = \boldsymbol{a} \times \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$

a_i and b_i are 16-bits unsigned fixed-point data, and 8 integer bits and 8 fraction bits.

Multiplier and adder are both 16-bits fixed point multiplier and adder.

2. System Architecture

2.1 No pipeline

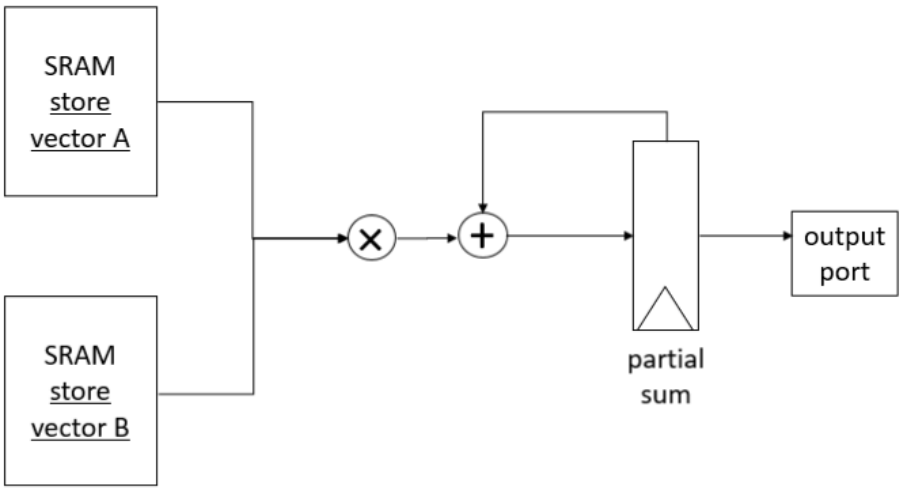

Fig.1 No pipeline

Step1: fetch a_i and b_i from SRAM

Step2: MAC operation, the multiplier-accumulator (MAC unit) is composed of one multiplier and adder, which is psum = psum + a×b.

2.2 Pipeline

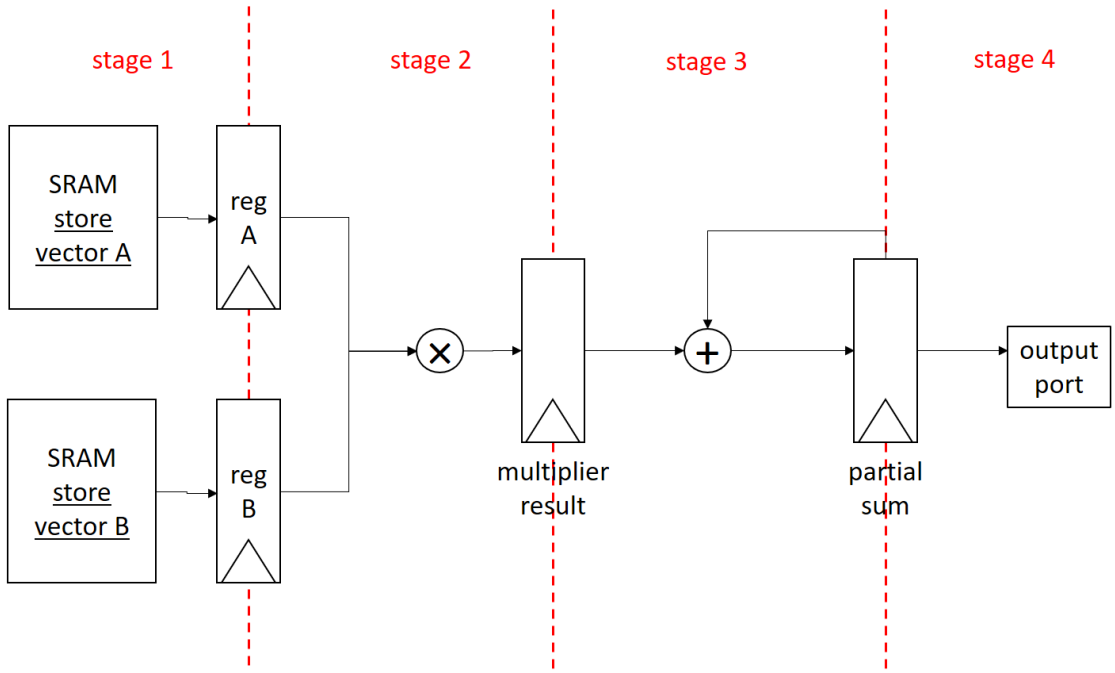

Fig.1

The system (fig.1) consists of 4 stages pipeline:

Stage 1: Fetch one data of vector A and vector B to regA and regB.

Stage 2: Multiply data stored in regA and regB

Stage 3: Add product of multiplier with previous partial sum: psum = psum + multiplier_result.

Stage 4: send the dot product result to output.

3. Lab Task

Task-1:

Implement the system in Fig.1 and Fig.2 to calculate the dot product of 2 vectors in no-pipeline and pipeline version:

A = [1.25, 2.5, 2.5, 2, 3, 2, 1.25, 3.5, 4.5, 2]

B = [1.5, 1.5, 5, 2, 5, 3, 3.5, 5, 4, 3]

Task-2:

Analysis your simulation result, and find at which time spot the system get the final result in both version and explain the reason clearly.

Task-3:

Follow the Design Compiler and Cadence encounter tutorial, complete the logic synthesis and Place & Route of your design.

4. Submit files

1. lab-report: containing simulation, synthesis and place&route screenshots.

    a) simulation: analyze your simulation waveform

    b) synthesis: compare no-pipeline and pipeline area, power and timing.

    c) P&R: screenshot your final chip layout

2. Verilog module code

3. testbench

4. Design compiler synthesis report: <your_design>_max_timing.repC, <your_design>_area.repC, <your_design>_power.repC