IDP

Fixed-Point Matrix Multiplication

Pamula Bhargav Ram
EE16BTECH11024
Rallabandi Rishideep Reddy
EE16BTECH11031

Problem statement and Implementation Strategy

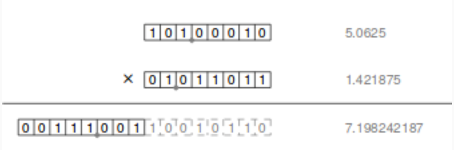
Table of contents

Introduction

Fixed Point Matrix Multiplication

The main idea of fixed point arithematic is to interpret bit words as integers coupled with a scale factor: $\frac{z}{2^n}$

Multiplication: The product of a $Q_{a,b}$ variable by a $Q_{c,d}$ yields a $Q_{a+b,c+d}$ variable.



Fixed Point Matrix Multiplication

Table 1: Fixed-Point Examples

Binary	Hex	Integer	Floating Point Fraction	Fixed-Point Fraction	Actual
0100000000000000	4000	16384	0.50000000	0.50000000	1/2
0010000000000000	2000	8192	0.25000000	0.25000000	1/4
0001000000000000	1000	4096	0.12500000	0.12500000	1/8
0000100000000000	0800	2048	0.06250000	0.06250000	1/16
0000010000000000	0400	1024	0.03125000	0.03125000	1/32
0000001000000000	0200	512	0.01562500	0.01562500	1/64
0000000100000000	0100	256	0.00781250	0.00781250	1/128
0000000010000000	0080	128	0.00390625	0.00390625	1/256
0000000001000000	0040	64	0.00195312	0.00195312	1/512
0000000000100000	0020	32	0.00097656	0.00097656	1/1024
0000000000010000	0010	16	0.00048828	0.00048828	1/2048
0000000000001000	0008	8	0.00024414	0.00024414	1/4096
0000000000000100	0004	4	0.00012207	0.00012207	1/8192
0000000000000010	0002	2	0.00006104	0.00006104	1/16384
0000000000000001	0001	1	0.00003052	0.00003052	1/32768
0010101010101011	2AAB	10923	0.33333000	0.33334351	0.33333
0101101001111111	5A7F	23167	0.70700000	0.70700073	0.707
0000000000001010	000A	10	0.0003141592	0.00030518	0.0003141592
0000000000000011	0003	3	0.000086476908	0.00009155	0.000086476908

Fixed Point Matrix Multiplication

It is similar to a normal matrix multiplication. Let us recall that inner product between two vectors.

$$\begin{bmatrix} A_x & A_y & A_z \end{bmatrix}\begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = A_xB_x + A_yB_y + A_zB_z = \vec{A}\cdot\vec{B}$$

From the figure it is clear that C_{ij} = inner-product of i^{th} row of matrix A and j^{th} column of matrix B.

$$c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + a_{14}b_{41}$$

$$\underset{2 \times 4}{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix}} \underset{4 \times 3}{\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix}} = \underset{2 \times 3}{\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}}$$

$$c_{22} = a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} + a_{24}b_{42}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}$$

Problem Statement

Problem Statement

- The input is sent through Arduino and then transmitted to the icoboard whenever the clock is triggered to positive.
- In order to flash verilog file in icoboard type make v_fname = mat_mul in terminal of RaspberryPi.
- Icoboard is flashed with verilog code which includes fixed point matrix multiplication logic.
- After executing the matrix multiplication, Icoboard outputs the matrix value to another Arduino where output is displayed.

Implementation Strategy

Implementation Strategy

- Arduino is used to input the matrix values. (i.e. matrix A and matrix B)
- Arduino converts floating point input to fixed point output.
- Ico board receives input from Arduino where matrix multiplication is programmed.
- The Project will be implemented for $(n \times n)$ matrix. And 8-bit fixed-point numbers will be used (n must be edited in the Arduino and verilog codes).
- For ease of demo, the implementation will be done for (2×2) matrix.
- Ico board sends the output after the multiplication process to another Arduino.
- The output matrix can be displayed in Arduino serial monitor after floating point conversion.

Implementation Strategy

- whenever the input is provided at the serial monitor of the input Arduino, then a clock signal is triggered to positive edge.
- every time the verilog code reads positive edge from the signal coming from the Arduino, the value of the input is stored in the registers.
- after all the required inputs have been provided, the code stops taking inputs from Arduino. and triggers a signal for Arduino to start the matrix multiplication logic.
- After the matrix multiplication is over, the values are stored in the registers.
- Later, when outputs are stored in registers, the code triggers another signal to the Arduino to start printing the output matrix.
- the output Arduino will trigger a signal for taking outputs stored in the registers at every clock signal.

References

References

1. download and setup RaspberryPi from: https://www.raspberrypi.org/downloads/raspbian/
2. get Arduino setup from: https://www.arduino.cc/en/Main/Software
3. Prorject codes are available in: https://github.com/BhargavasRamus/IDP_FPGA_Spring19/tree/master/codes

THANK YOU