CS2323 Computer Architecture
HW-4

Shreyas Jayant Havaldar
CS18BTECH11042

This document is generated by LaTeX

October 18, 2019

1

The generalized formula for array multiplication(of n x n matrix) using systolic array is obtained to be:

$$(3n) - 2 \tag{Ans}$$

This can be understood by realizing that $b_{0,0}$ will take **n** steps to reach from row 0, to row n-1. When $b_{0,0}$ would have reached the n-1th row, $a_{n-1,0}$ would have entered 0th column with it. This $a_{n-1,0}$ will take **n-1** further steps to reach the n-1th column. When $a_{n-1,0}$ will reach n-1th column, $a_{n-1,n-1}$ would have entered the 0th column. This $a_{n-1,n-1}$ will then take **n-1** steps to reach the last column. Thus a total of n + n-1 + n-1 = **3n-2** steps are required to perform the entire multiplication.

2

```
for(row_big=0; row_big<R; row_big = row_big + B) //B is the blocking factor
for(col =0; col<C; col++)
for(to=0; to<M; to++)
for(ti =0; ti <N; ti++)
for(i =0; i<K; i++)
for(j =0; j<K; j++)
for(row=row_big; row_big<min(row_big+B,R); row++)
Output_fmaps[to][row][col] += Weights[to][ti][i][]j] * Input_fmaps[ti][S*row+i][S*col+j]
```

3

```
__device__ void addFunc1(int *a, int *b, int *c)

__global__ void addFunc2(int *a, int *b, int *c)

__host__ void random_ints(int* x, int size)

__host__ int main(void)
```

4

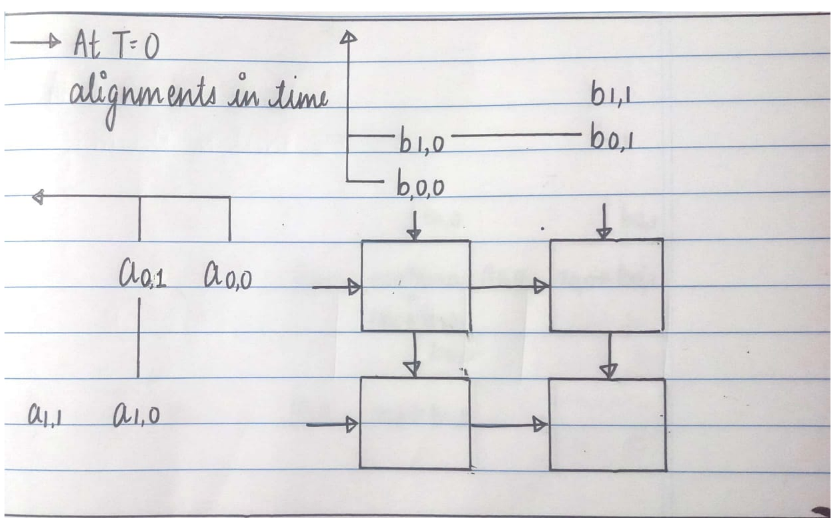
Variables	Location
x_{dim}	Register
y_{dim}	Register
iteration	Register
pqr	Local
ABC	Global
maxValue	Global

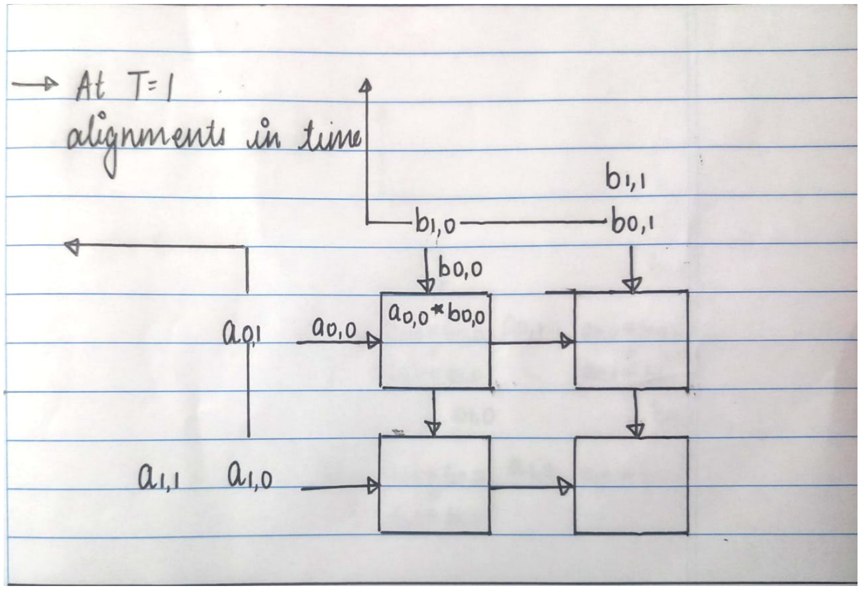
5 **(a)** Dimension of the matrix is 16 x 16. Cache size is 128 bytes.
(b)

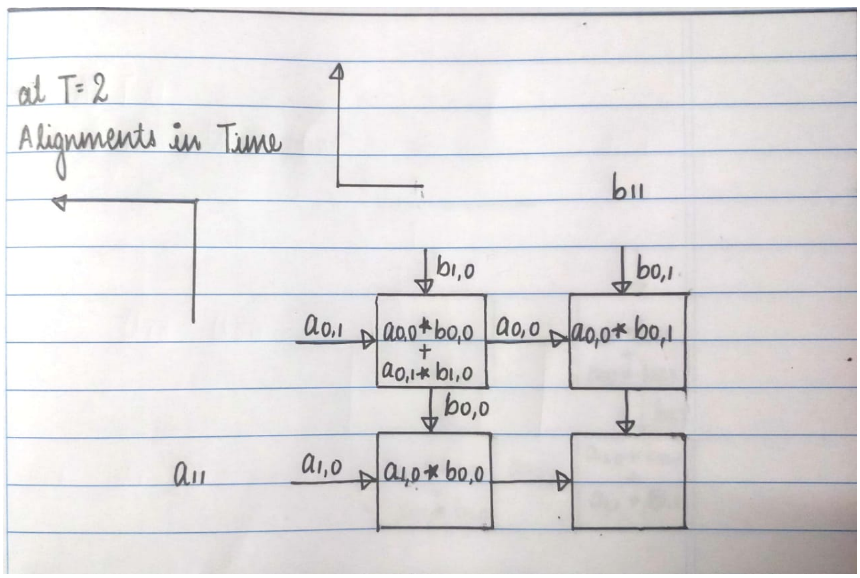
Unblocked Cache:
Hits: 192
Misses: 320
Input Matrix Misses = 64 | Output Matrix Misses = 256

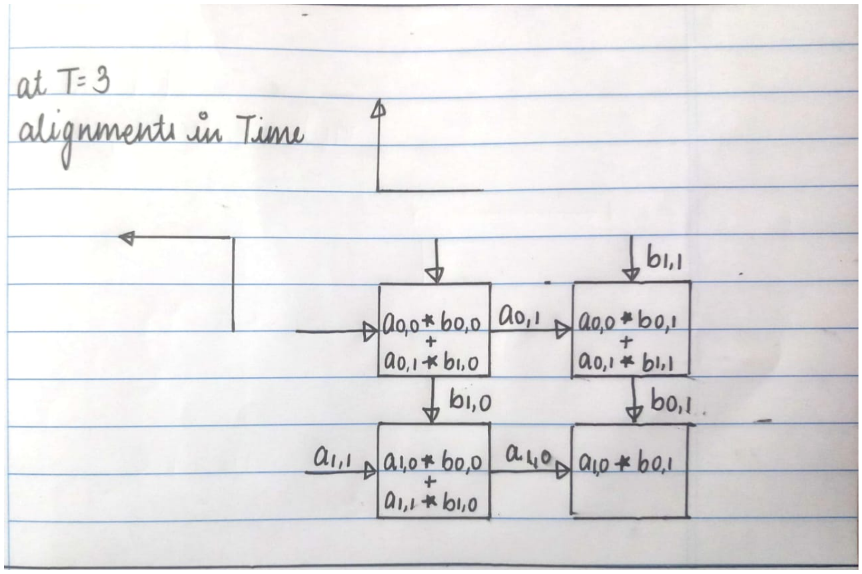
Blocked Cache:
Hits: 384
Misses: 128
Input Matrix Misses = 64 | Output Matrix Misses = 64

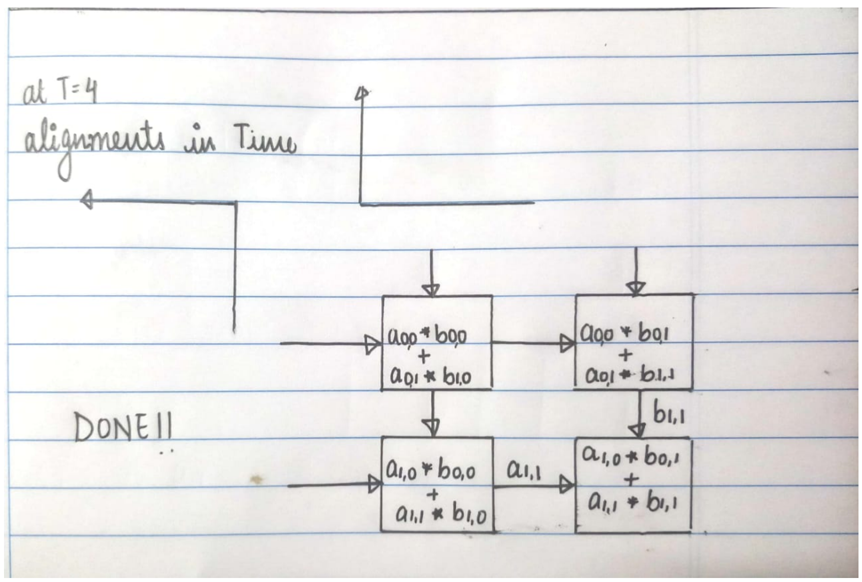
6 Shown are the 5 stages for $\mathbf{A} \cdot \mathbf{B}$:




At T=1
alignments in time
b1,1
b1,0
b0,1
b0,0
a0,0*b0,0
a0,1
a0,0
a1,1
a1,0


at T=2
Alignments in Time
b11
b1,0
b0,1
a0,1
a0,0*b0,0 + a0,1*b1,0
a0,0
a0,0*b0,1
b0,0
a11
a1,0
a1,0*b0,0


at T=3
alignments in Time
b1,1
a0,0 * b0,0 + a0,1 * b1,0
a0,1
a0,0 * b0,1 + a0,1 * b1,1
b1,0
b0,1
a1,1
a1,0 * b0,0 + a1,1 * b1,0
a1,0
a1,0 * b0,1


at T=4
alignments in Time
a0,0 * b0,0 + a0,1 * b1,0
a0,0 * b0,1 + a0,1 * b1,1
b1,1
DONE!!
a1,0 * b0,0 + a1,1 * b1,0
a1,1
a1,0 * b0,1 + a1,1 * b1,1

7 **(a)** 4
(b) 5.5
(c) 5.6875
(d) 5.6953125

8

Instruction	Semantics
v.ld vr1, 20[r1]	vr1 $\leftarrow$ ($[r1+20], [r1+28]$)

Note: 8byte doubles stored.

---

9 We know: AI $= \dfrac{TotalFLOPS}{TotalDRAMBytes}$
Note: Assuming 8 byte double elements in all the matrices.

(a)
Case 1:
Total Flops $= n^2$; Total DRAM Bytes $= 3n^2 \times 8$

$$AI = \frac{n^2}{3n^2 \times 8} = 1/24 \qquad \text{(Ans)}$$

Case 2:
Total Flops $= n^2/4$; Total DRAM Bytes $= 3n^2 \times 8$

$$AI = \frac{n^2/4}{3n^2 \times 8} = 1/96 \qquad \text{(Ans)}$$

(b)
Case 2:
Total Flops $= n^2/4$; Total DRAM Bytes $= 3n^2/4 \times 8$

$$AI = \frac{n^2/4}{3n^2/4 \times 8} = 1/24 \qquad \text{(Ans)}$$

---

10 We know: AI $= \dfrac{TotalFLOPS}{TotalDRAMBytes}$
Note: Taking 1GB $= 10^9$ Bytes and 1MB $= 10^6$ Bytes wherever required.

(a)
Performance of ZU19EG FPGA obtained: $\dfrac{75}{100} \times 66TOPS = 49.5TOPS$
OPS to classify 1 image $= 1.5GOPS$

$$\therefore TotalImagesClasssifiedIn1Second = \frac{49.5TOPS}{1.5GOPS} = \frac{49500}{1.5} = \mathbf{33000} \qquad \text{(Ans)}$$

(b)

8bit fixed:

Total Flops = $1.5GFLOPS$; Total DRAM Bytes = $50MB$

$$AI = \frac{1.5G}{50M} = \mathbf{30} \tag{Ans}$$

Binarized:

Total Flops = $1.5GFLOPS$; Total DRAM Bytes = $7.4MB$

$$AI = \frac{1.5G}{7.4M} = \mathbf{202.702} \tag{Ans}$$

---

11 We know: Peak FLOP = AI × Memory Bandwidth

Note: Taking 1GB = 10^9 Bytes and 1MB = 10^6 Bytes wherever required.

Peak FLOP = 2199GFLOP/s

$\therefore AI = \dfrac{PeakFLOP}{MemoryBandwidth}$

MCDRAM

$$AI = \frac{2199GFLOP/s}{372GB/s} = \mathbf{5.9112} \tag{Ans}$$

DRAM

$$AI = \frac{2199GFLOP/s}{77GB/s} = \mathbf{28.558} \tag{Ans}$$
