FPGAs, HLS Tools & Runtime Systems

(Super)Advisors: Frederic Desprez, Francois Broquedis, Olivier Muller

Georgios Christodoulis

CORSE−LIG

gchristodoulis@gmail.com

Overview

FPGA

Description

A *Field Programmable Gate Array* is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "Field-programmable".

FPGAs are semiconductor devices that are based around a matrix of configurable logic blocks (BLEs) connected via programmable interconnects.

FPGAs Structure

LUT

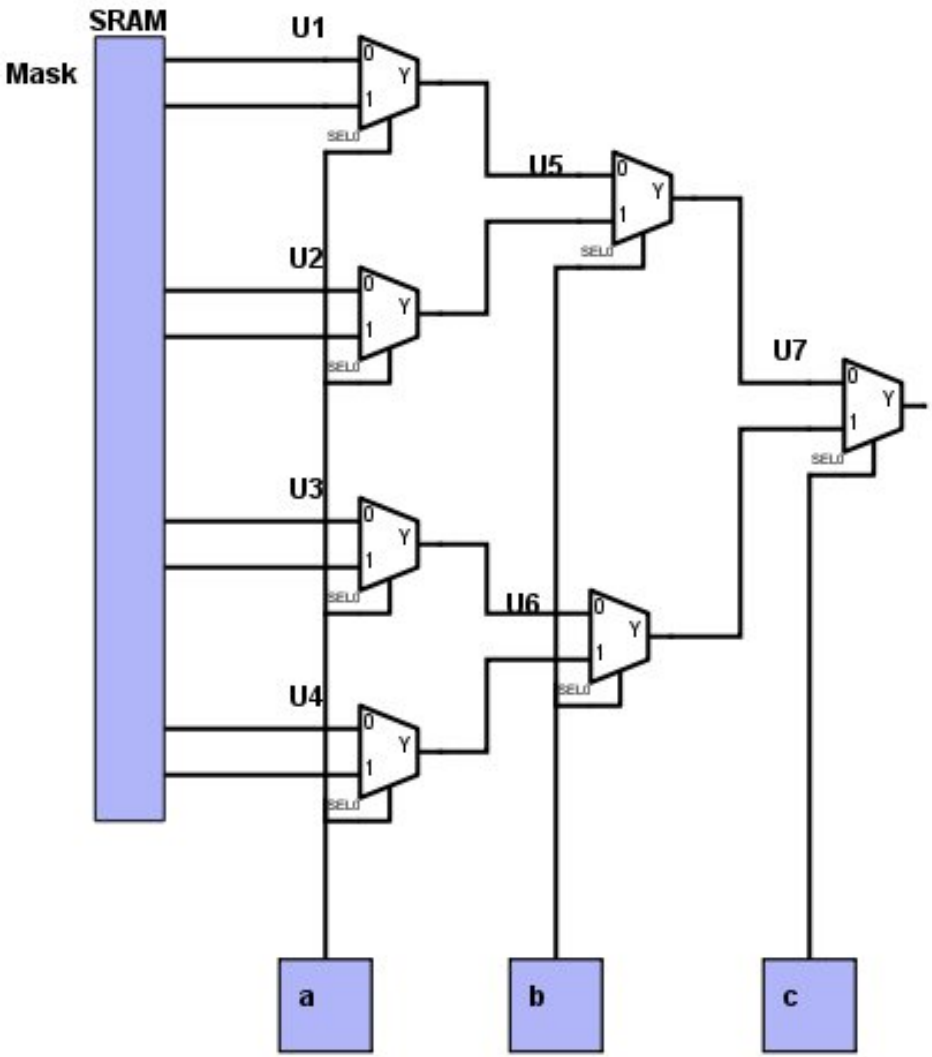


Figure: 3 stages of 2x1 MUX

- It is a table that ditermines what the output is for any given input
- A state-less interconnection of any number of gates (no feedback loops)
- Implemented *multiplexing* a combination of SRAM bits

FPGAs Structure

LUT Example

$$y = (a + b) \cdot c$$

a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

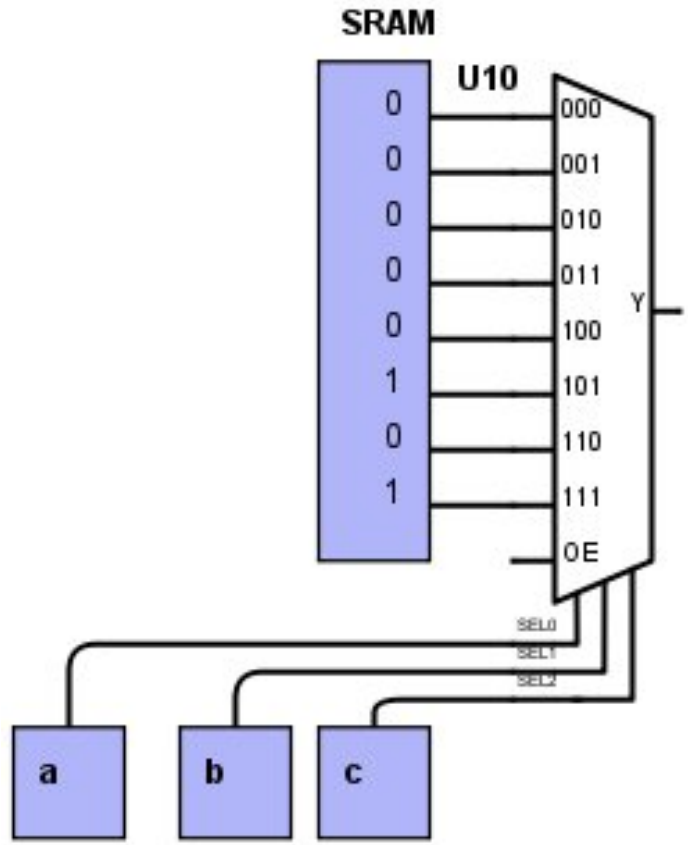


Figure: $y = (a + b) \cdot c$

FPGAs structure

BLE

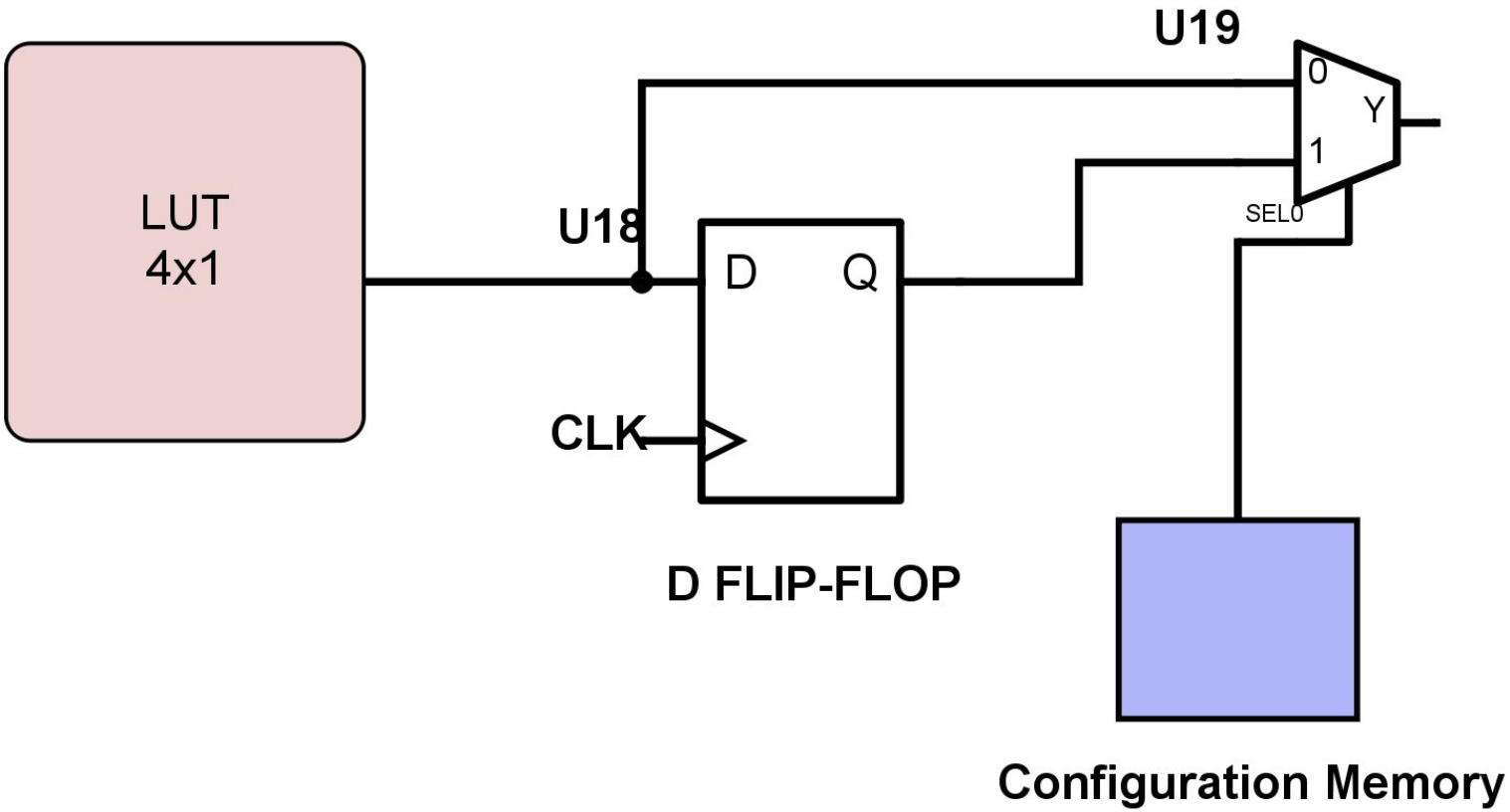


Figure: Basic Logic Element

FPGAs structure

Overview

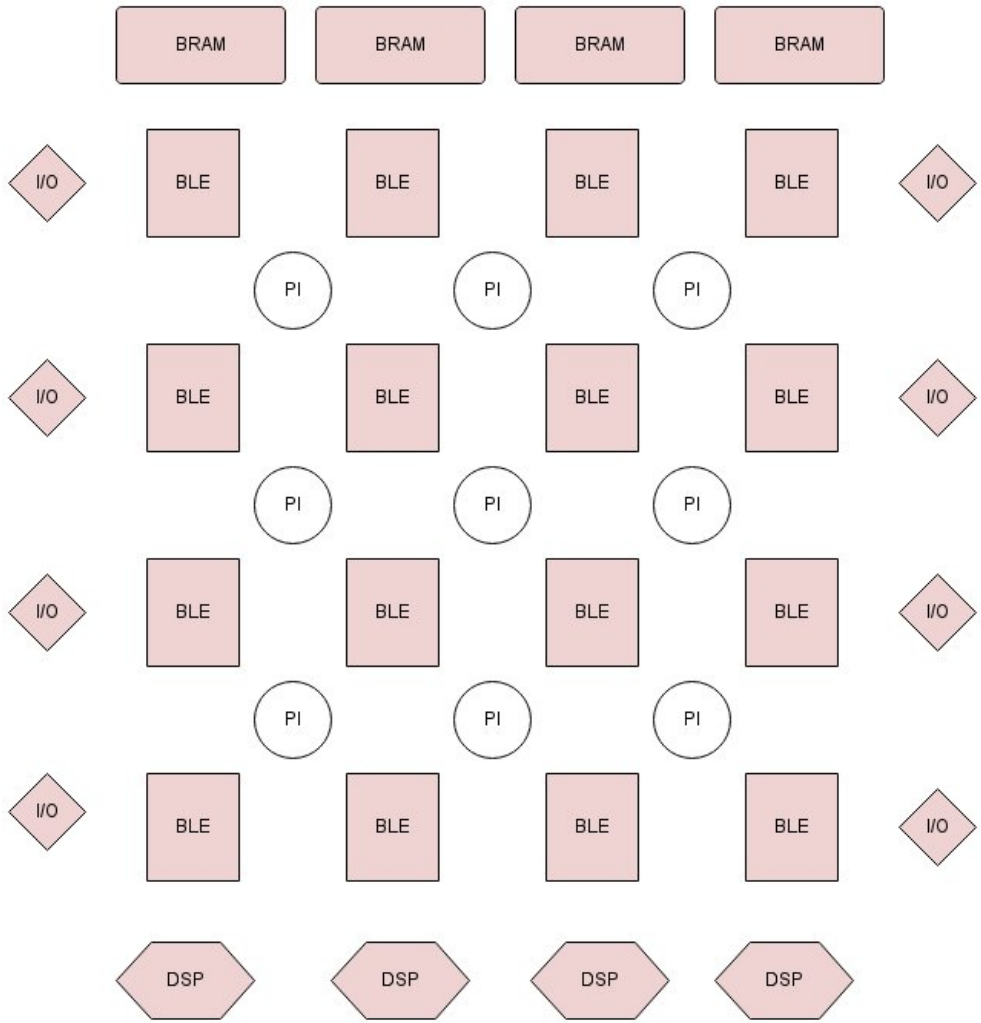


Figure: FPGAs Complete Overview

Problem Description

Matrix Multiplication

$$C = A * B$$

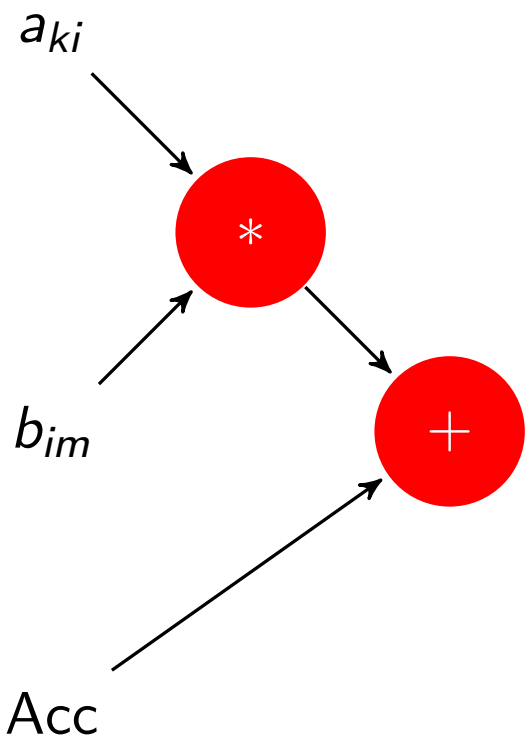
$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

$$\begin{vmatrix} c_{11} & \dots & c_{1n} \\ \vdots & c_{km} & \vdots \\ c_{n1} & \dots & c_{nn} \end{vmatrix} \quad \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kn} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} \quad \begin{vmatrix} b_{11} & \dots & b_{1m} & \dots & b_{1n} \\ b_{21} & \dots & b_{2m} & \dots & b_{2n} \\ \vdots & & \vdots & & \vdots \\ b_{n1} & \dots & b_{nm} & \dots & b_{nn} \end{vmatrix}$$

No Directives

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

$$c_{km} = a_{k1} b_{1m} + a_{k2} b_{2m} + \cdots + a_{kn} b_{nm}$$



- $2n$ operations $\forall$ element
- n^2 elements

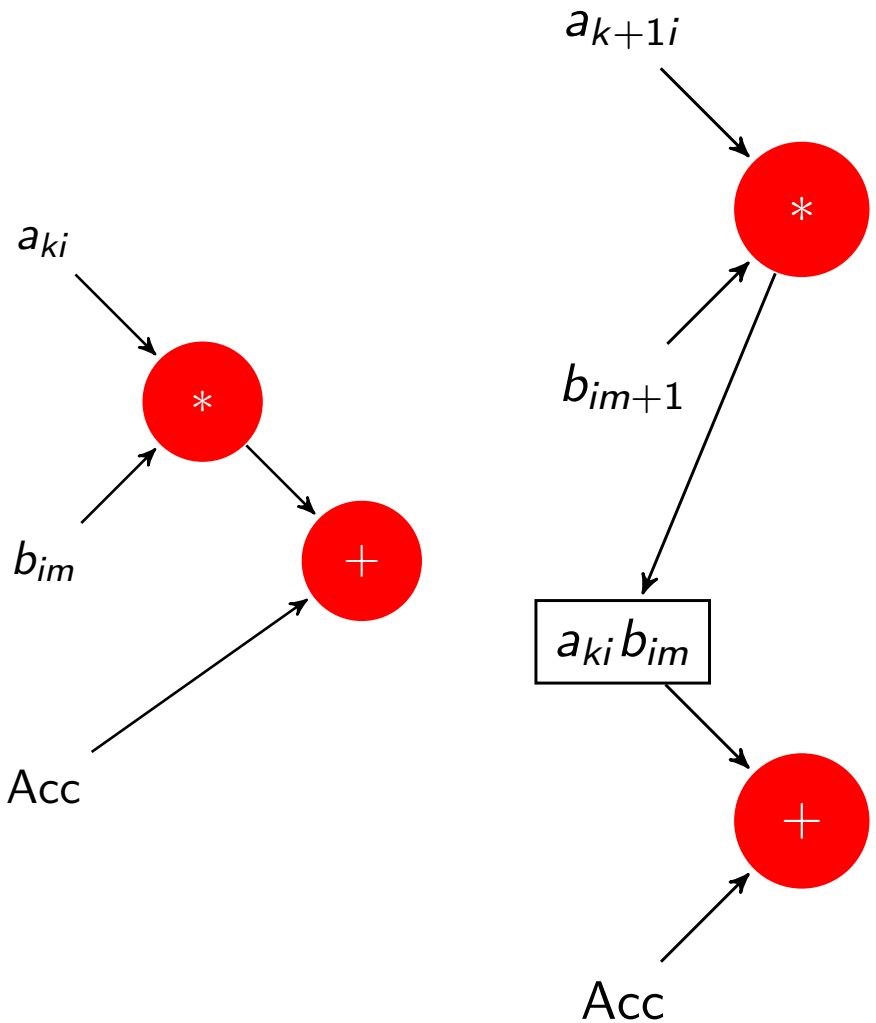
$\Rightarrow$ $2n^3$ operations

- 19MFLOPS on the tested platform

Inner Loop Unrolling

Opt1: Sum Mul Overlaping

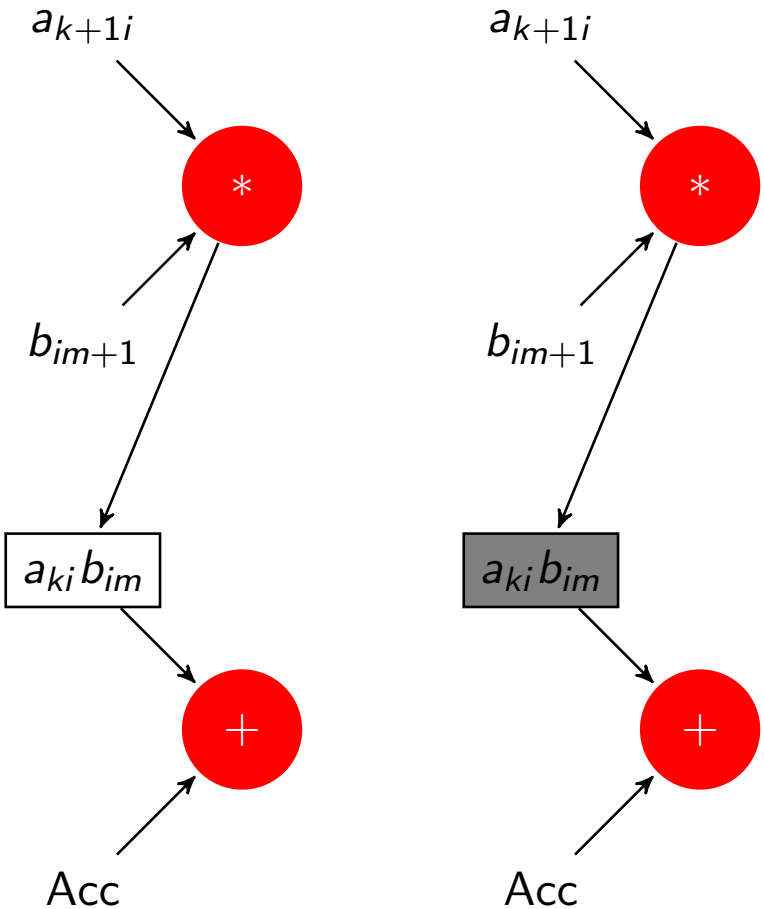
Our reference time interval is defined by the slowest operation which is the multiplication.

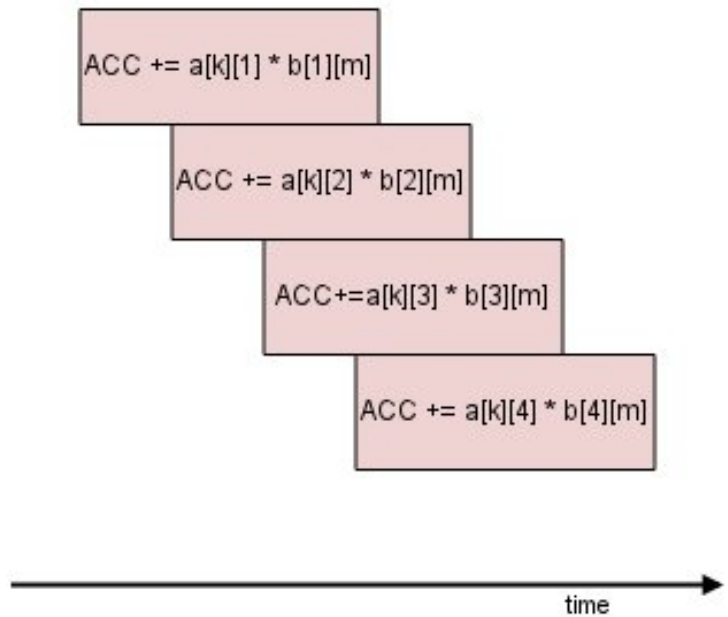


- Complete overlaping between Multiplication and addition
- 35MFLOPS on the given machine

Pipeline

Initiation Interval is called the number of cycles between two new iterations.
In this case it is indicated by the time that the addition register is occupied.



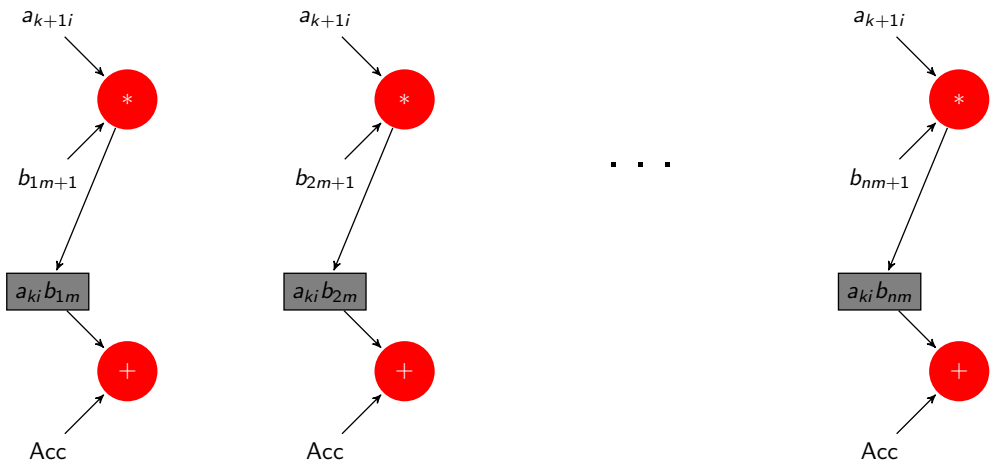


J-loop unrolling

$$a_{k1} \quad a_{k2} \quad \dots \quad a_{kn}$$

$$\begin{matrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{matrix} \quad \begin{matrix} b_{12} \\ b_{22} \\ \vdots \\ b_{n2} \end{matrix} \quad \begin{matrix} \cdots \\ \cdots \\ \\ \cdots \end{matrix} \quad \begin{matrix} b_{1n} \\ b_{2n} \\ \vdots \\ b_{nn} \end{matrix}$$

- Extra Hardware
- Real Parallelism (Superscalar CPU architectures)
- Expected scaling: $\times n$
- Actual scaling: $\times 2$



Memmory Bounds: Dual Channeled memory $\Longrightarrow$ only 2 concurent operations.

Row / Col Partitioning

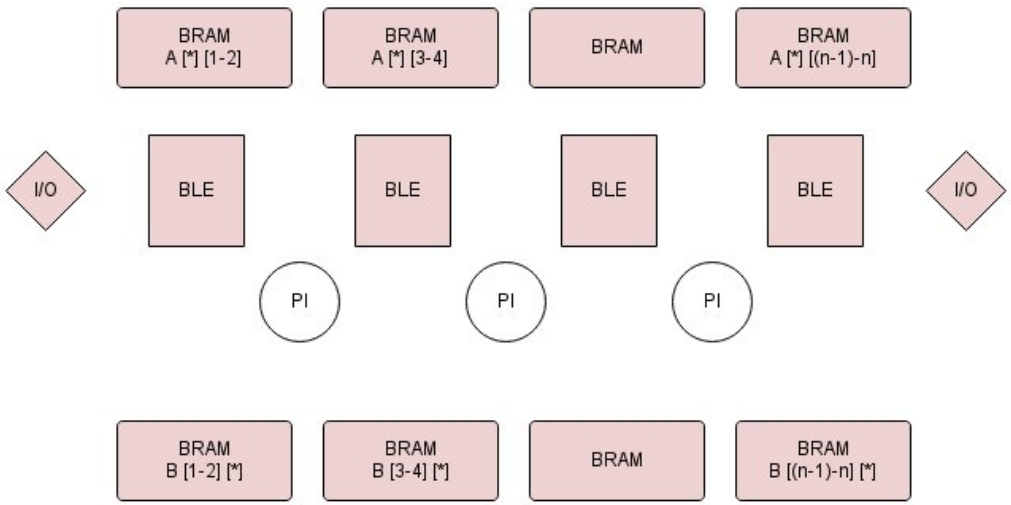


Figure: Distribute the array into multiple BRAMS