POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

Project Logic Brain : a Binary Neural Networks acceleration

Presented by:

Alexandre Riviello

Nathan Heraief

Mohammad Hossein Askari

Introduction

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Context:

- Goal: Accelerate a neural network inspired algorithm using a hardware/software solution.
- The SocKit board must be used.
- Two potential situations: A 5 minute case and a 24 hour case.

Presentation Plan

POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

Structure:

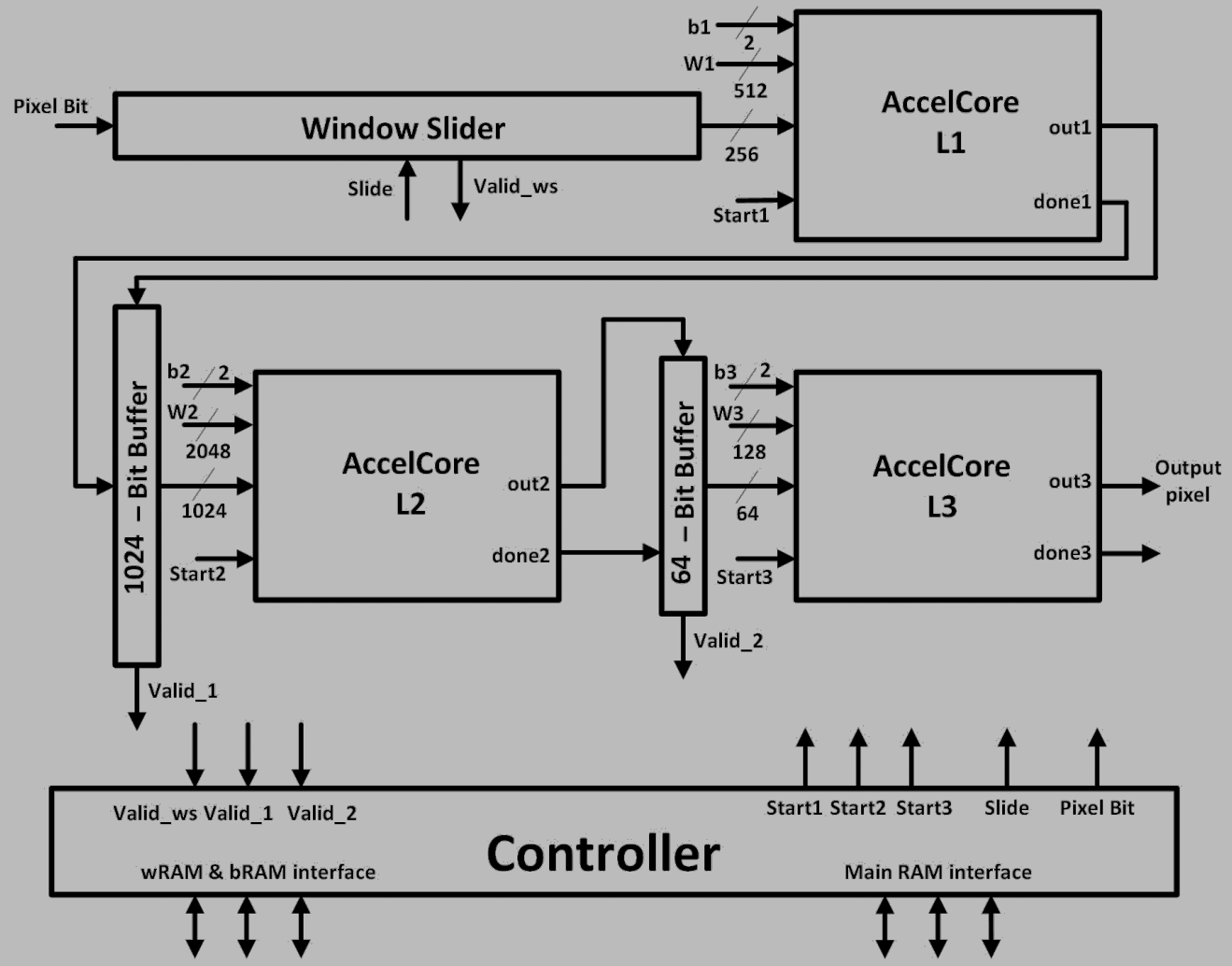
1. Architecture and features
   - NeuralCore
   - Memory Management
   - Controller Unit
2. Simulation and Verification
   - Test benches & scripts
   - Sniffer module
3. Results, specifications, problems & improvements
4. Conclusion

1. Architecture and Features

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Full Architecture:

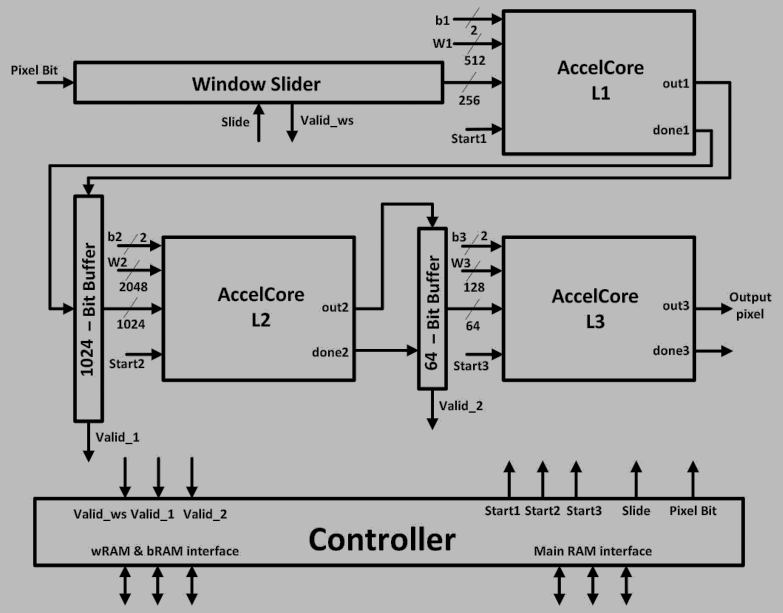
- Window Slider
- 3x AccelCore
- 3x Bit Buffer
- Controller
- Memory

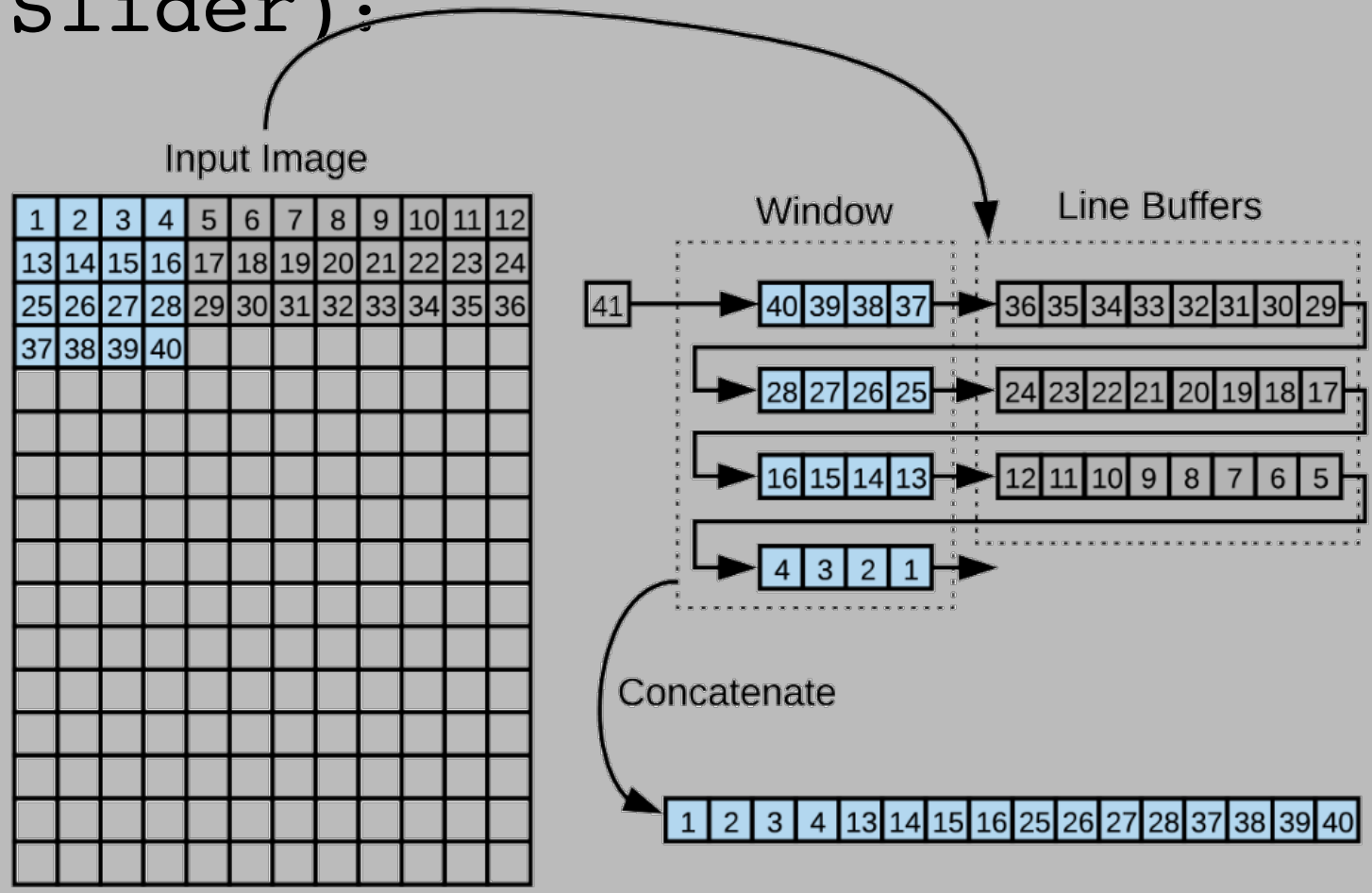


1. Architecture and Features

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

NeuralCore (Window Slider):



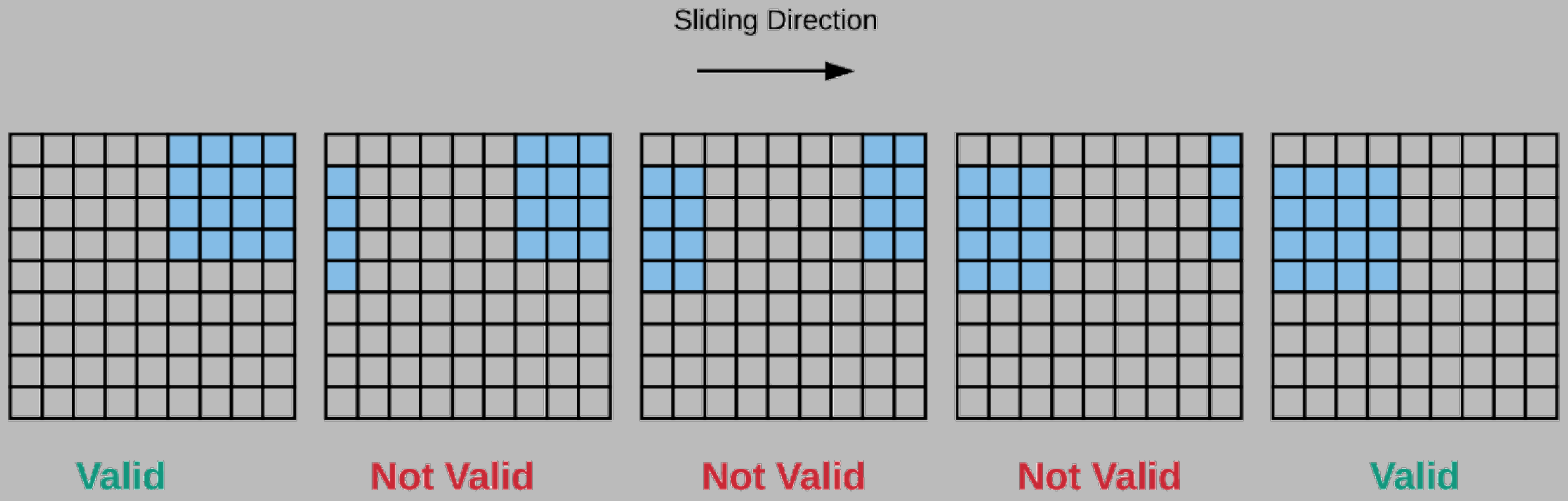


Inspired by Farabet, C., Poulet, C., Han, J. Y., & LeCun, Y. (2009, August). Cnp: An fpga-based processor for convolutional networks. In *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on* (pp. 32-37). IEEE.

1. Architecture and Features

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

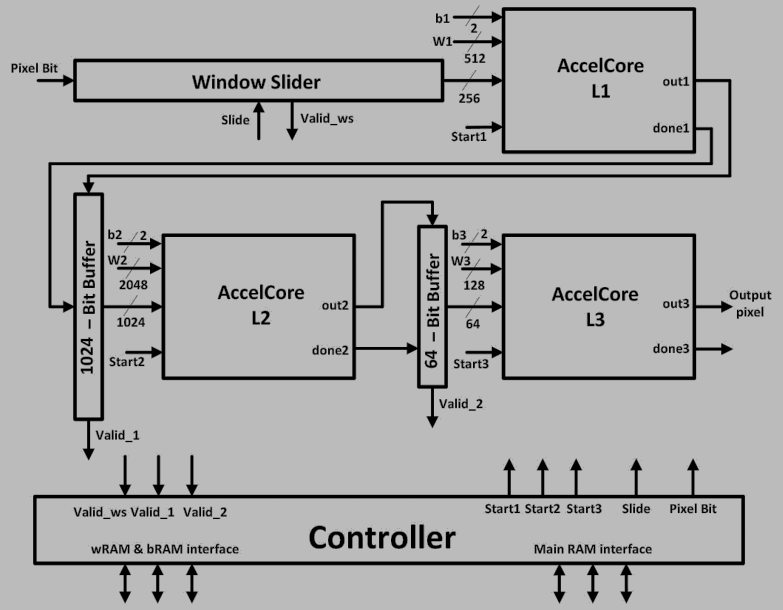
NeuralCore (Window Slider – part 2):

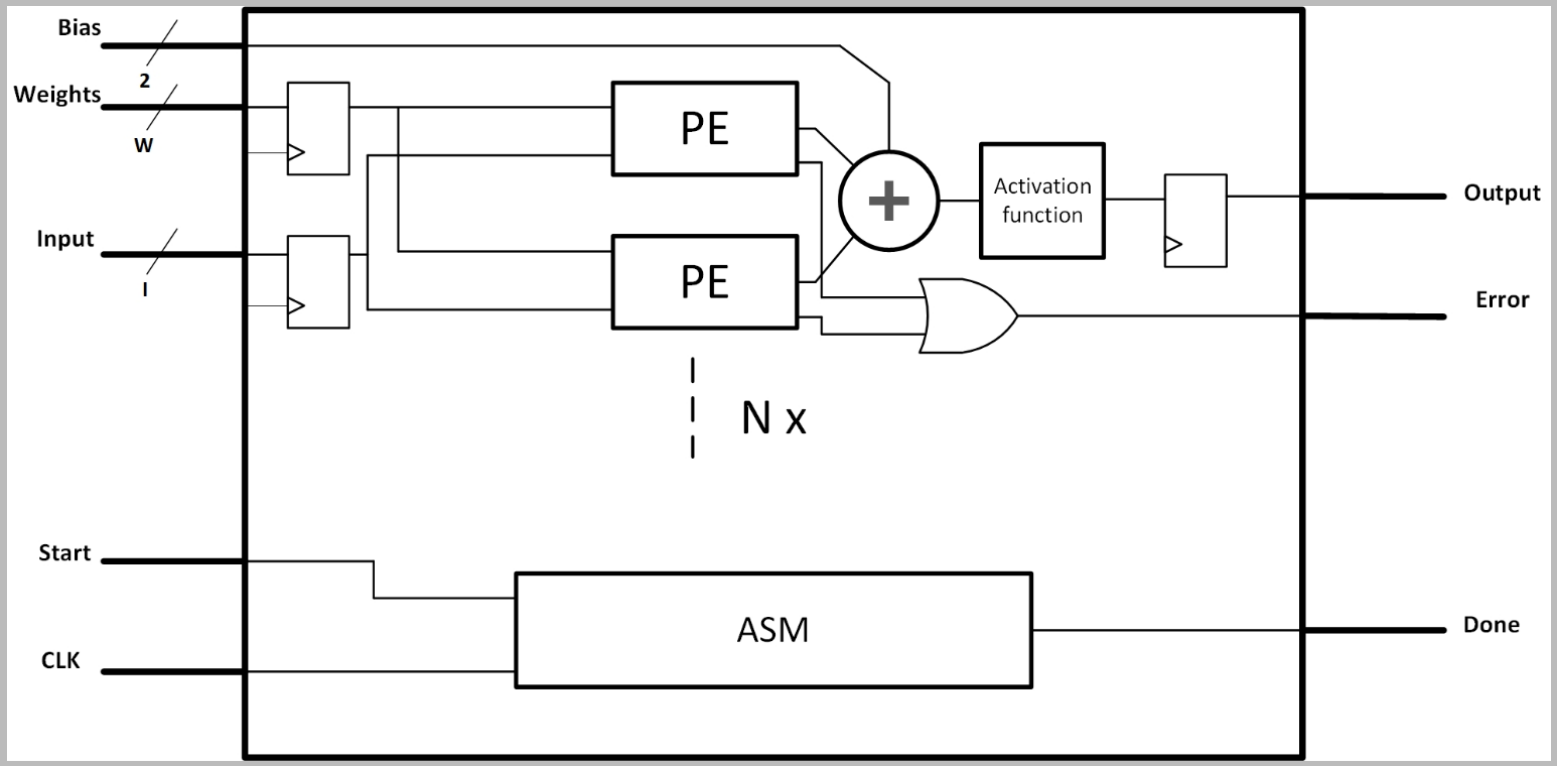


1. Architecture and Features

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

NeuralCore (AccelCore):

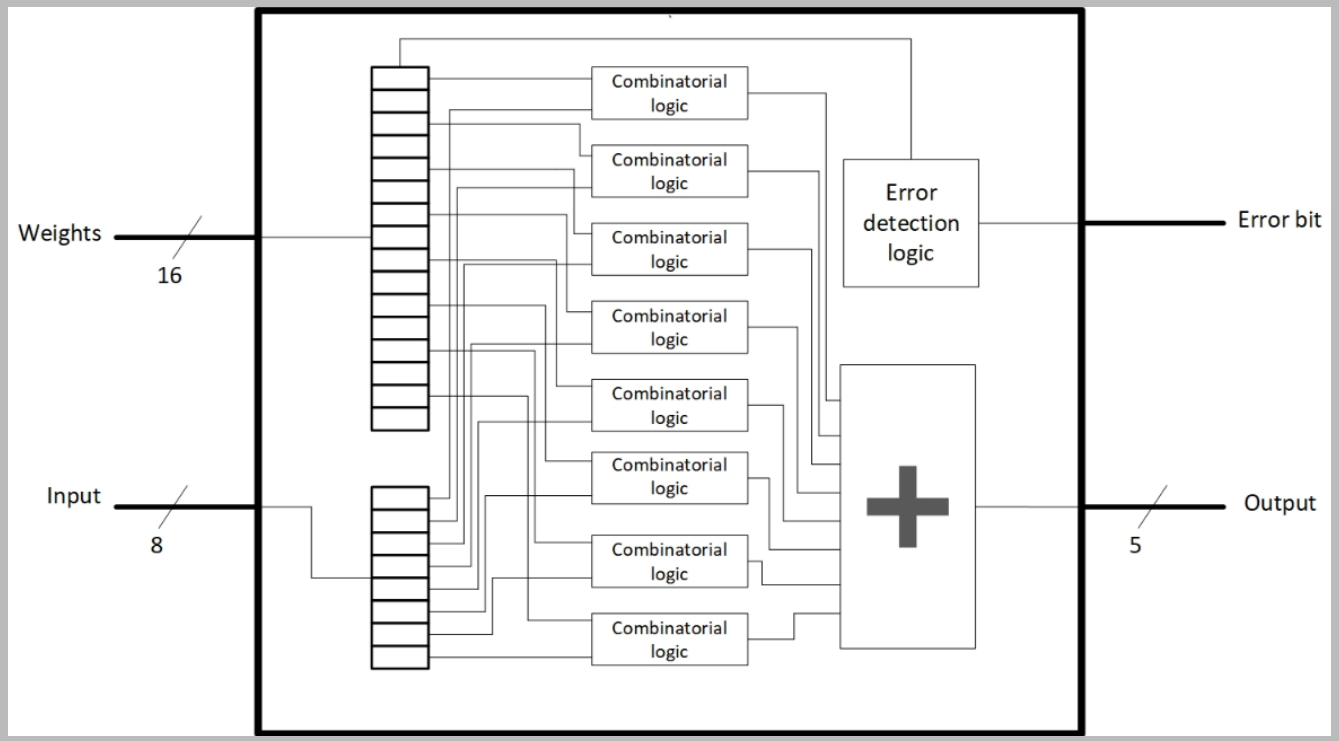


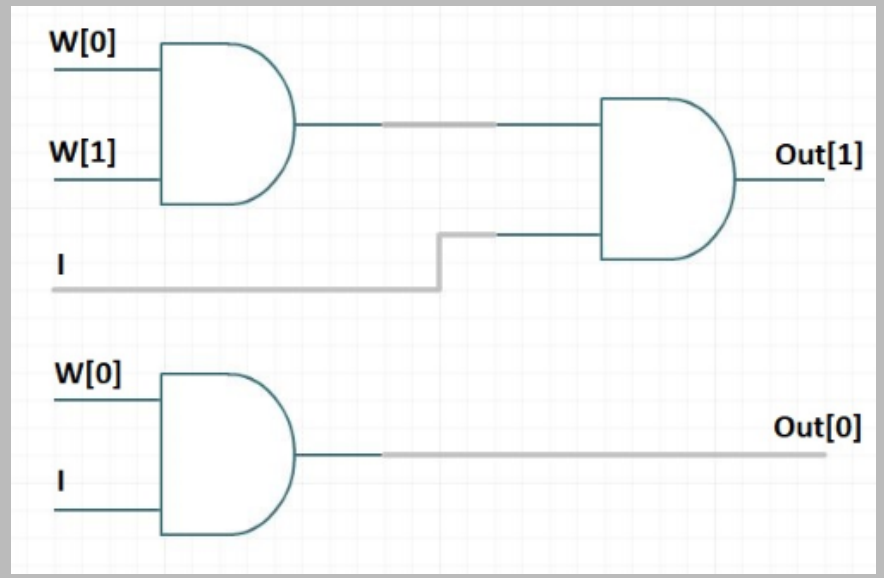


1. Architecture and Features

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

NeuralCore (AccelCore – part 2):

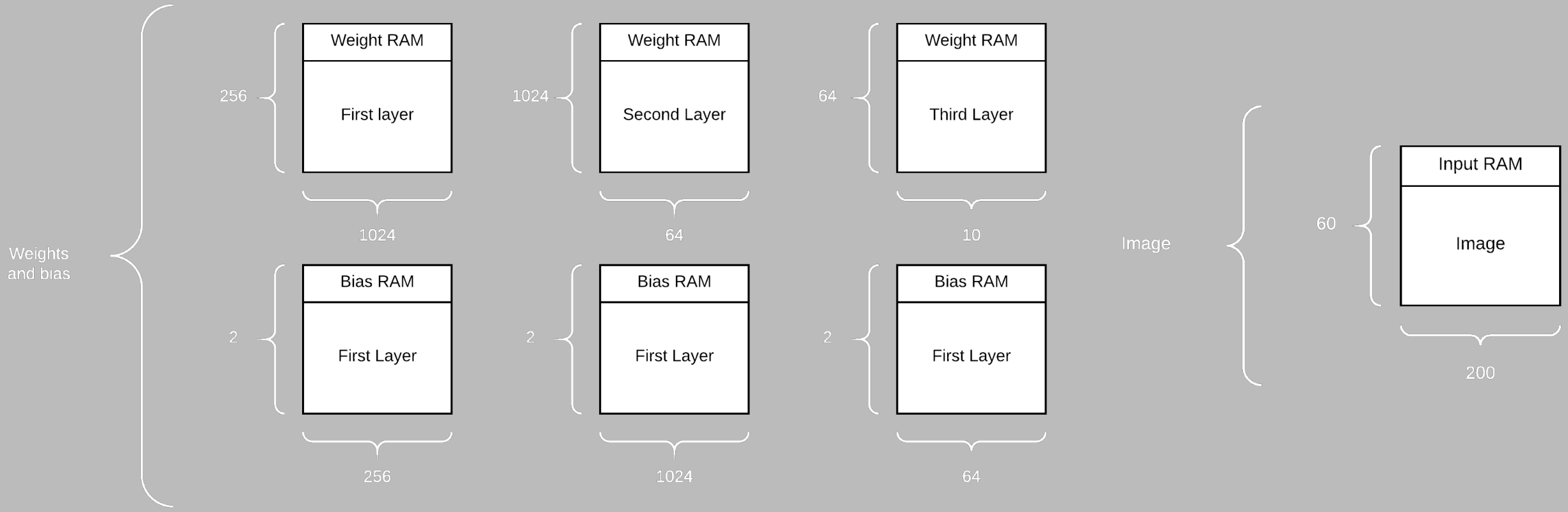




1. Architecture and Features

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Memory Management:



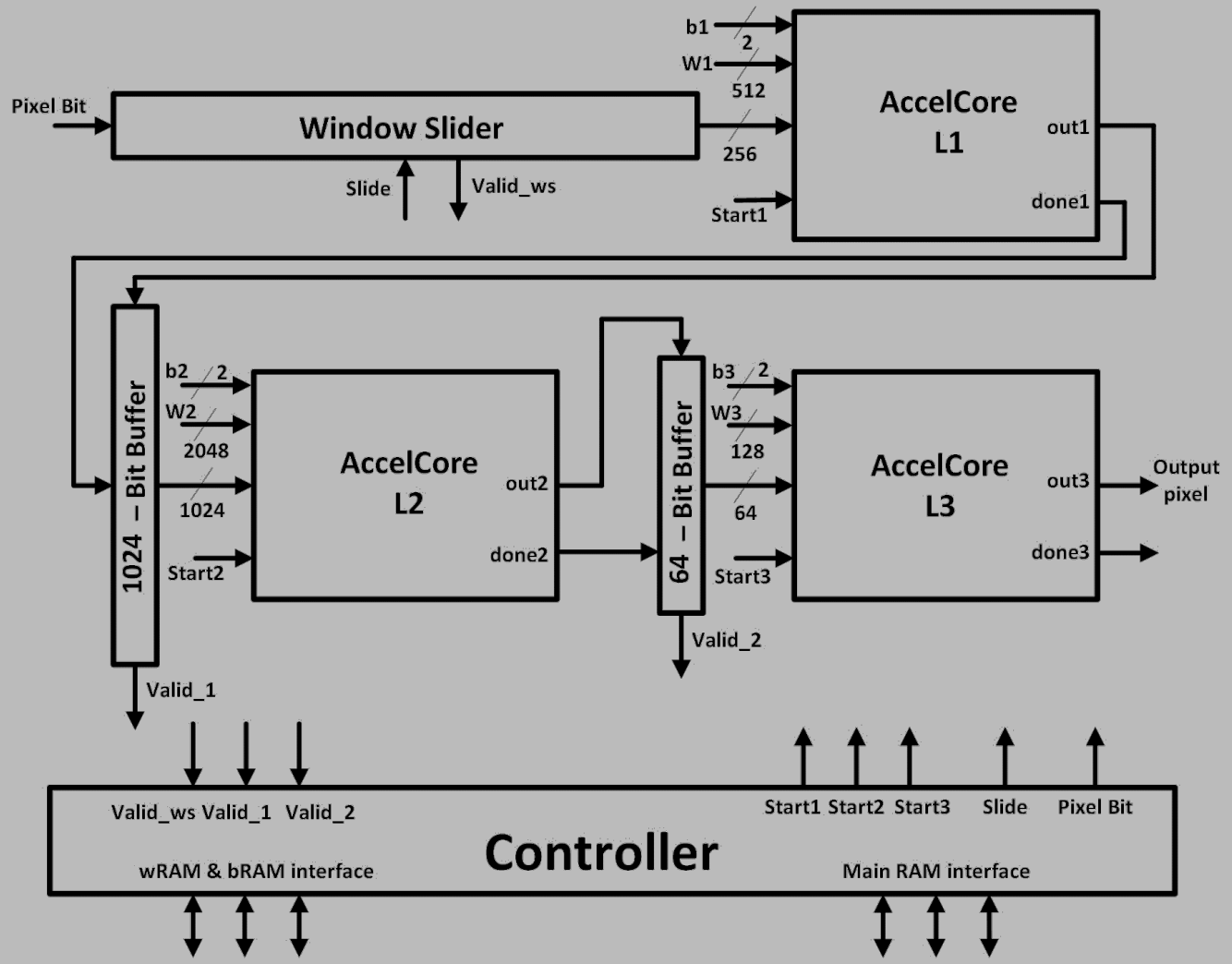
https://www.intel.com/content/www/us/en/programmable/quartushelp/13.1/mergedProjects/reference/glossary/def_m10k.htm

1. Architecture and Features

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Controller Unit:

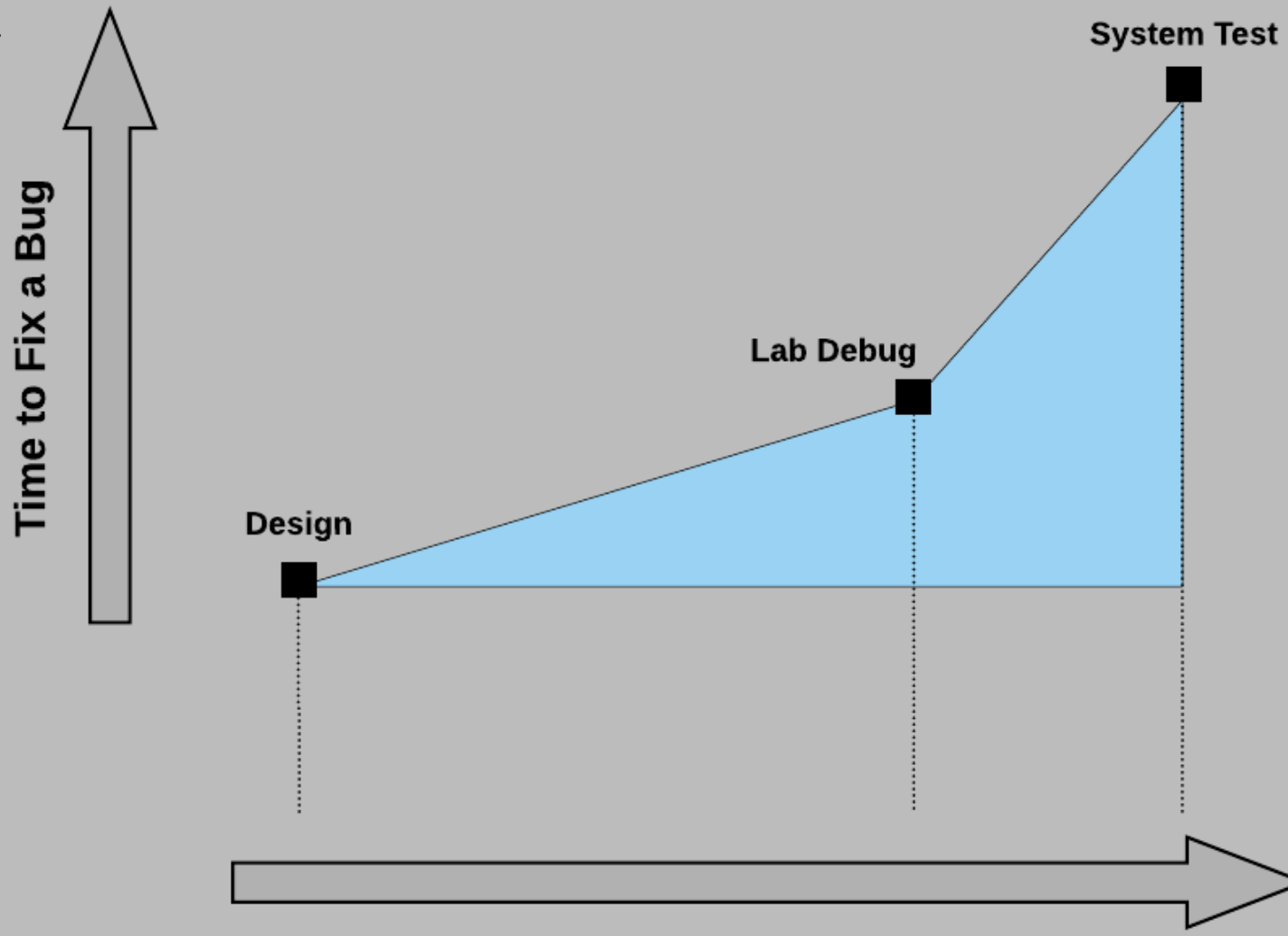
- Controller waits for 3 signals (valid_ws, valid_1 and valid_2)
- Once these signals are high, it reads all the RAM addresses consecutively within a same layer.



2.Simulation and Verification

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

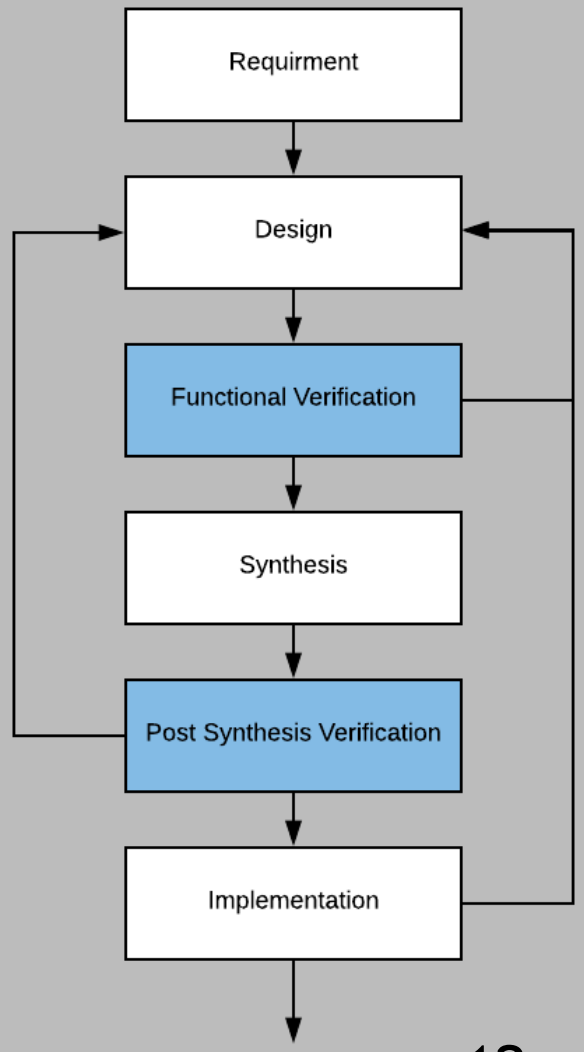
Cost (time and money) of fixing a bug over time (image from internsix)



2.Simulation and Verification

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

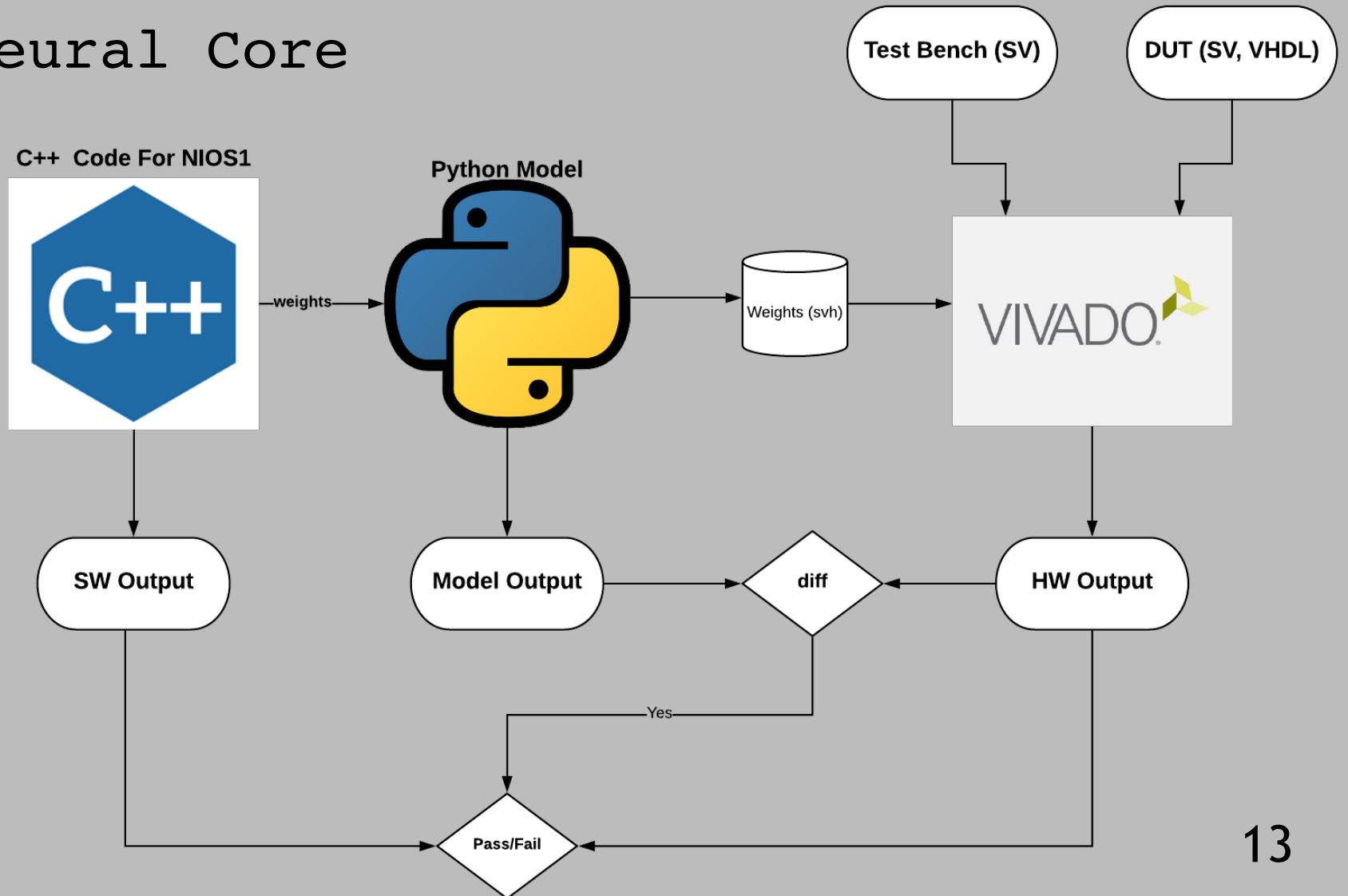
1. Verification and validation in FPGA design is a crucial step.
2. We splitted the verification into following steps:
    1. Writing test bench for every module (unit test)
    2. Writing System level test bench to verify the integration
    3. Writing software models to test the simulation results



2.Simulation and Verification

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Verification of Neural Core



2.Simulation and Verification

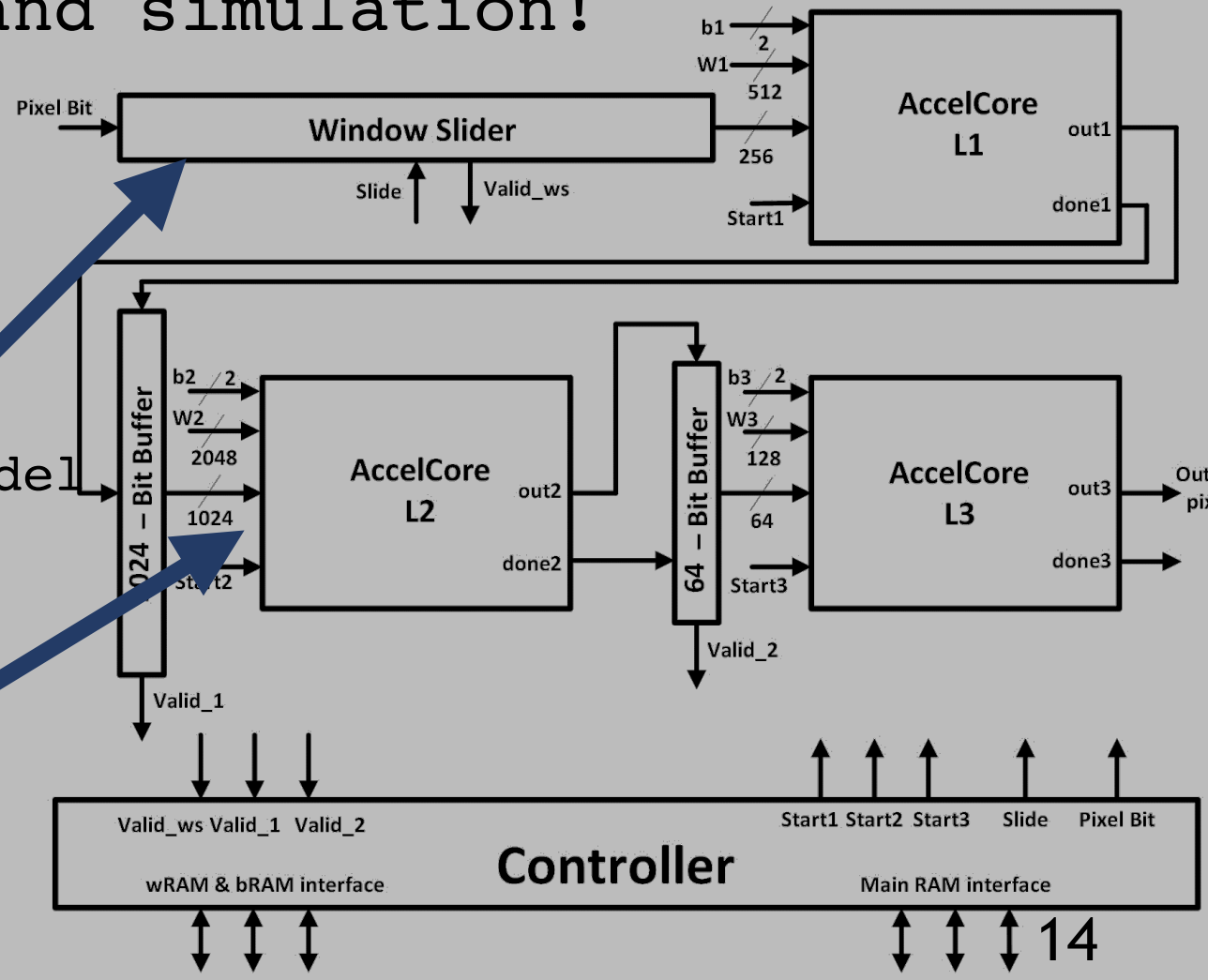
POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

We did a lot of scripting and simulation!

- Window Slider
- Accel Core
- Neural Core
- VGA output displayer
- Test and Simulation harness around Vivado

Python model

Python model

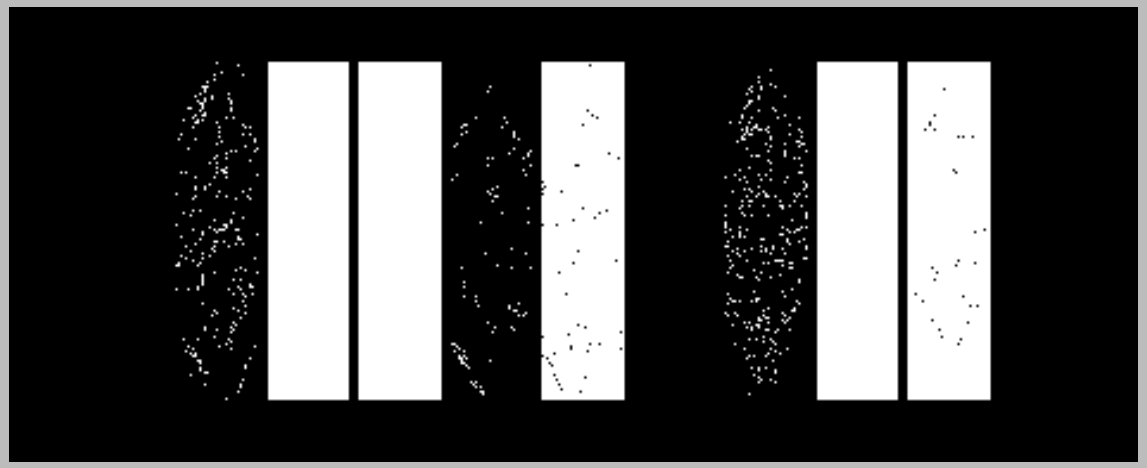


2.Simulation and Verification

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Our Software model generates exactly the same output!

40 x 40 x 10

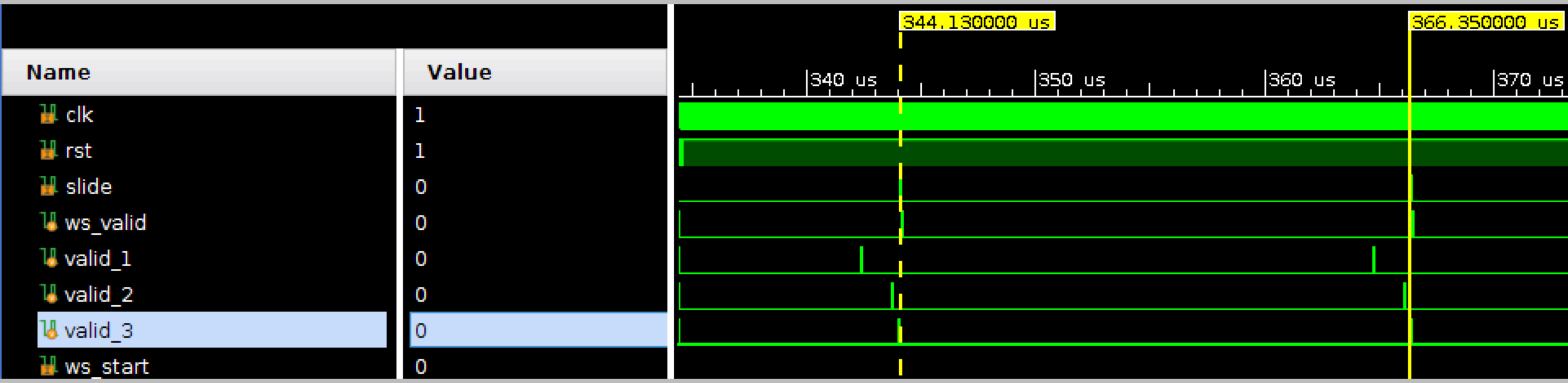


1024 x 60 x 10

2.Simulation and Verification

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

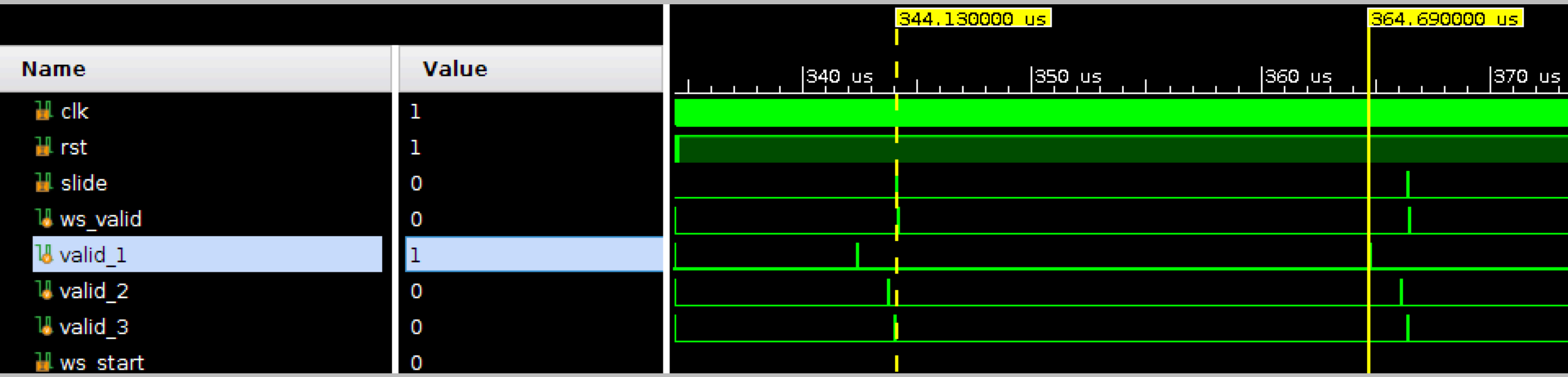
Our simulation shows that we can produce 8325 (184*45) pixels in 185ms @ 50MHz (22us per output)



2.Simulation and Verification

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

By adding a pipeline in window slider we can achieve around 10% boost



2.Simulation and Verification

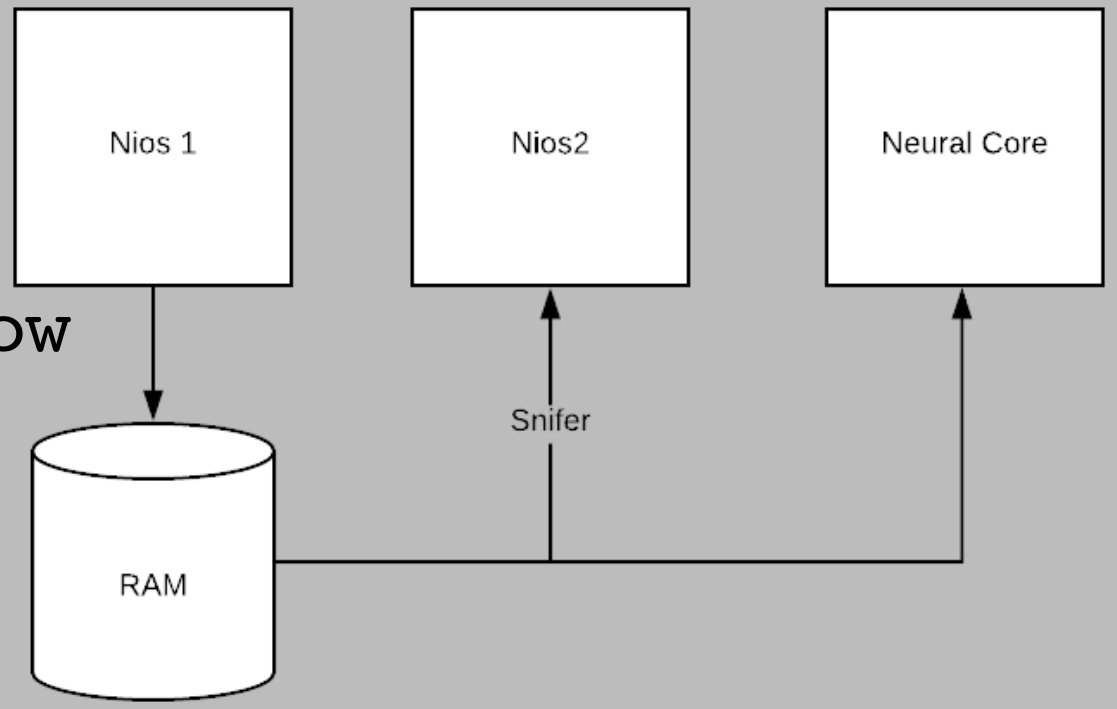
POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Demo!

2.Simulation and Verification

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

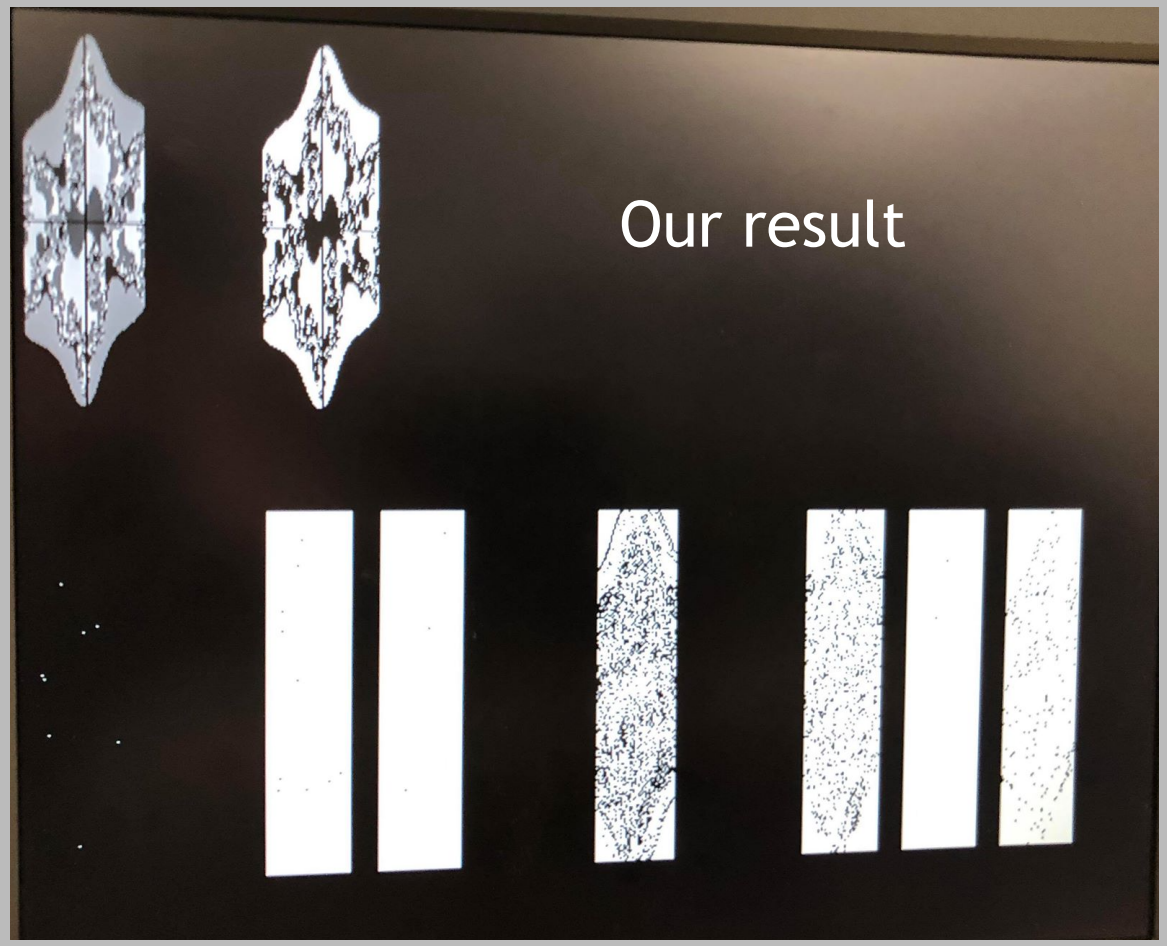
For validation, we used the second Nios as a packet Sniffer to report all transaction to the ram. This helped us to know what values are being written to the RAM module



3.Results/Specs/Problems & Solution

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

1. Results



3.Results/Specs/Problems & Solution

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

2. Specs : Timing

Time to load weights and bias:

335ms

Time to calculate worst scenario

256ms

3.Results/Specs/Problems & Solution

2. Specs : Resource consumed

Logic utilization (in ALMs) :

15,707 / 41,910 => (37 %)

Total registers :

23762

Total block memory bits :

3,472,880 / 5,662,720 => (61 %)

Total RAM Blocks :

459 / 553 => (83 %)

3. Problems and Solution

Problem	Solution	Implementation
Input can have different size	Preprocess the input	Zero Padding the input to max size (200x60)
Image is flipped	Check MSBs and LSBs	Flip all line buffers in neural core

3.Results/Specs/Problems & Solution

4. Futur Improvements

- Pipelining the Neural Core => 10%
- And ?

4. Conclusion

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Because figures are better than words :

2k times faster