מס' פרויקט:

20-1-1-2187

שם הפרויקט:

מימוש וסימולציה של מאיץ למערכות לומדות על רכיב FPGA

מגישים:

שי צבר 208723627

חיים גרודה 312562721

מאושר 16/01/21

יוני צייפרט

מנחה:

יוני זייפרט

מקום ביצוע:

אוניברסיטה

CNN – Convolutional Neural Network

- Achieved great success in image classification, speech recognition and more.
- Research hotspot in many scientific fields.
- Widely used in the industry, such as autonomous robot vision.

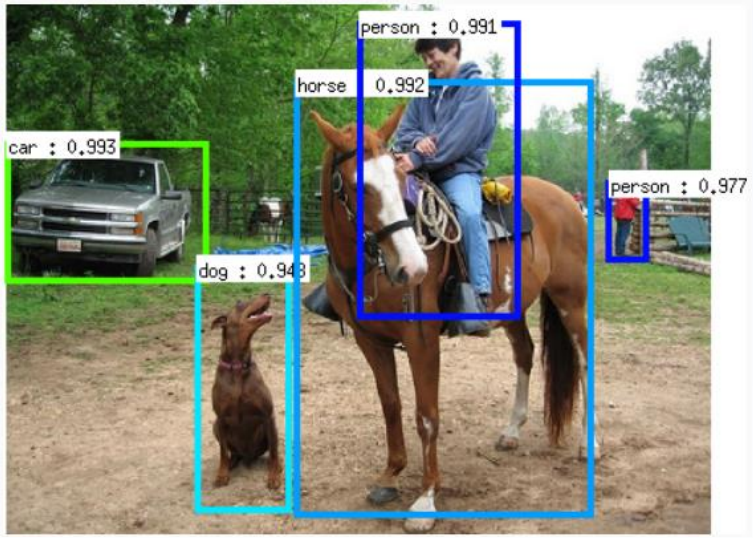


Figure: image recognition illustration [google]

- **CNN Structure**
  - **Convolution -** Feature extraction using filters
  - **Pooling -** Reduce spatial dimensions
  - **Fully Connected -** Final classification decision

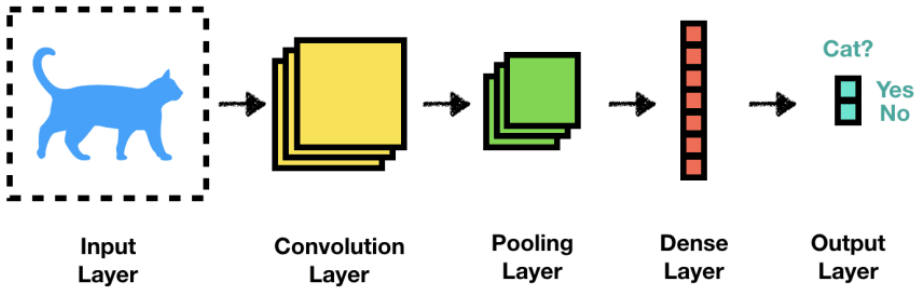


Figure: CNN layers [google]

CNN Acceleration

- **Convolutional layers:**
  - Computationally intensive
  - **85%-90%** of computations [source]
  - SW – sequential calculations using nested loops
  - HW – parallel calculations
- **Pooling**
  - Reduce spatial dimensions by taking Max/Avg
- **Fully Connected**
  - Matrix multiplications
  - Memory Heavy

$$Out_o = \sum_{i=1}^{Ni} In_i \times W_{io} + Bias_o$$

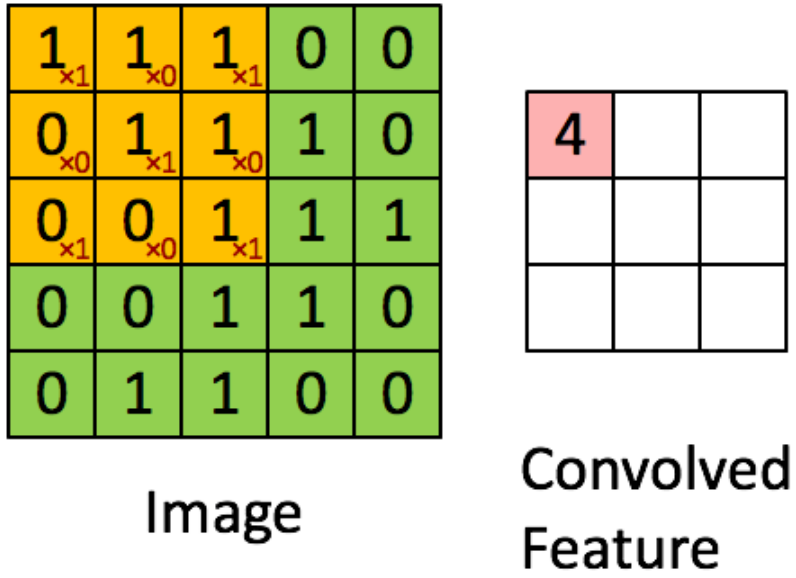


Figure: 2D convolution. [source]

System Requirements

- **Fixed Point Calculations:**
  - Better computation time
  - Less hardware resources
  - Lower accuracy

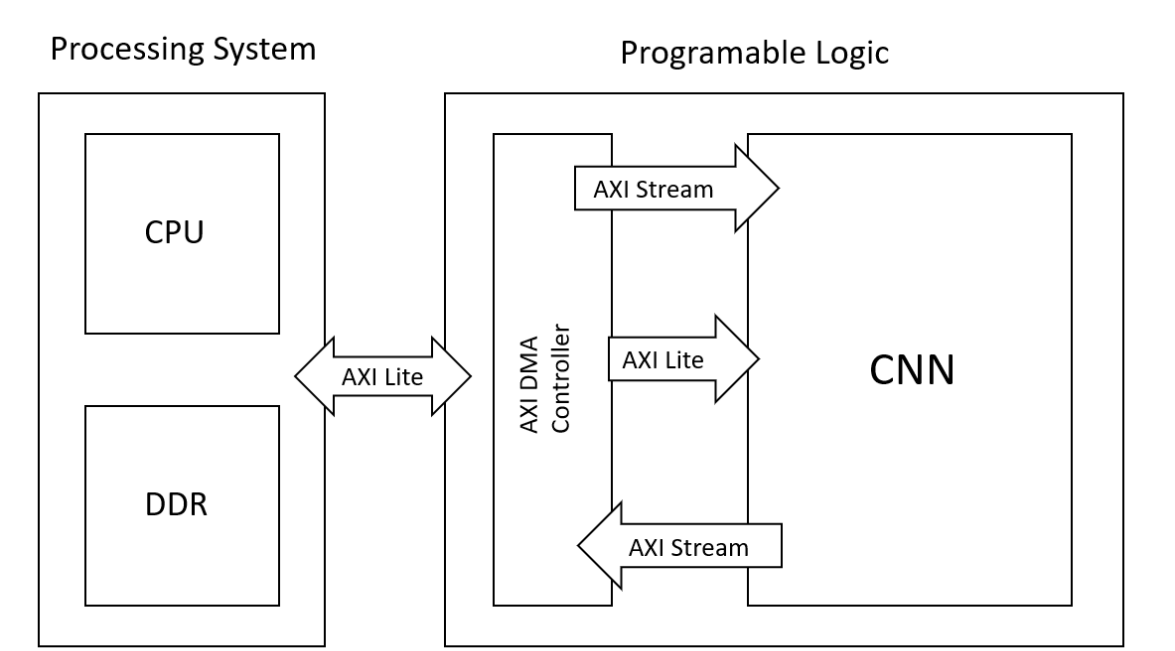
- **High Level Synthesis (HLS)**
  - RTL abstraction
  - Efficiently build and verify hardware
  - Better control over optimization of design architecture

- **Flexible Implementation:**
  - Adjustable for different CNN models
  - Varying Kernel size, Image size, number of network layers
  - Based on FPGA limits

- **Simulation:**
  - Test CNN core operation
  - Time comparison with SW
  - Accuracy comparison with SW
  - SW will run on PS and laptop

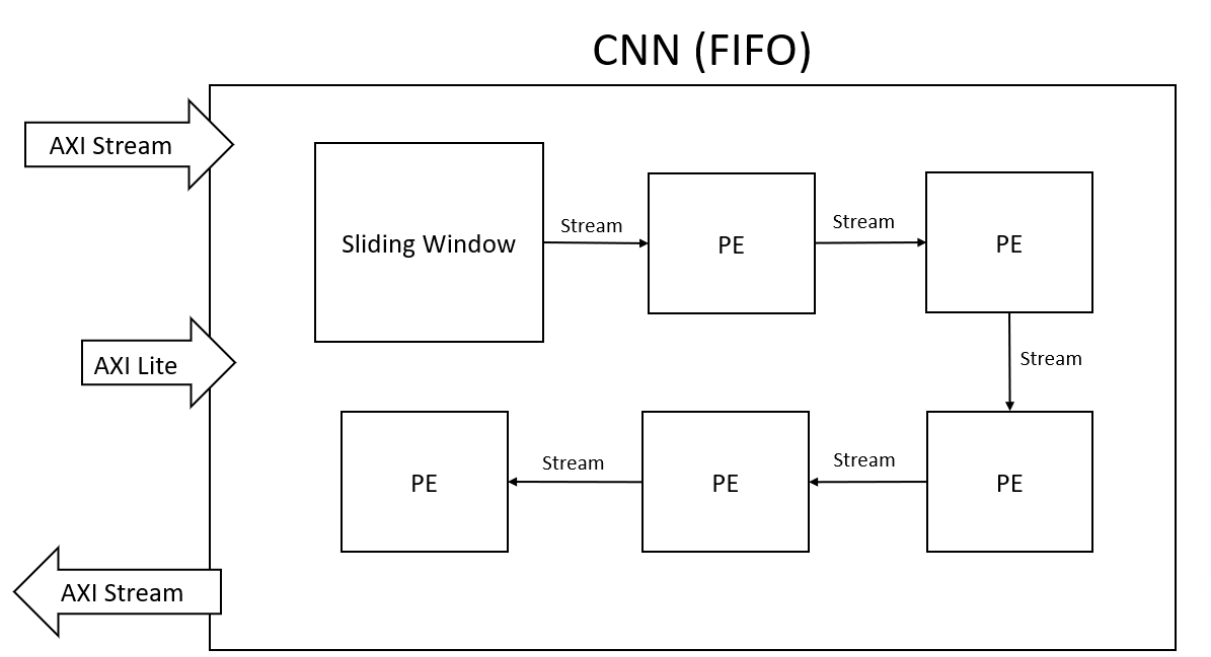
Design - Top

- **Processing system (PS):**
  - CPU – Input, output, control and configuration
  - DDR – memory unit
- **Programmable Logic (PL):**
  - AXI DMA controller
  - CNN core
- **AXI Stream\Lite interfaces**
  - Stream – high throughput (input data)
  - Lite – static data (kernel/weights)



Design – CNN Core

- **FIFO implementation**
  - Data is processed “on the fly”
  - No PL-PS bottleneck
  - AXI Stream for high throughput of processed data
  - AXI Lite for configuration / static data
  - FPGA resources can be utilized for more CNN layers, or for a bigger sliding window



Design – Sliding Window

Sliding Window Block

- Convert image section $(k \times k)$ into vector
- Data reuse – pixel transferred only once
- Block holds $(k-1)n + k$ pixels, held inside $(k-1)$ FIFO buffers of size $(n-k)$, and k^2 registers

n – Image Dimension

k – Kernel Dimension

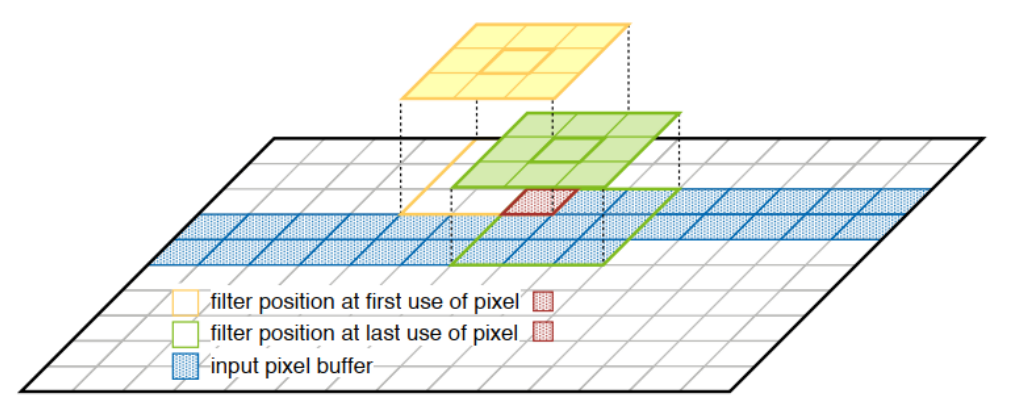


Figure: Data reuse / sliding window illustration [source]

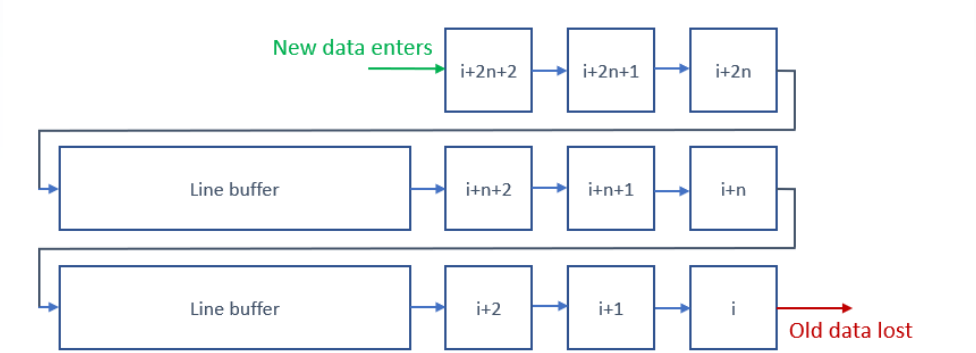
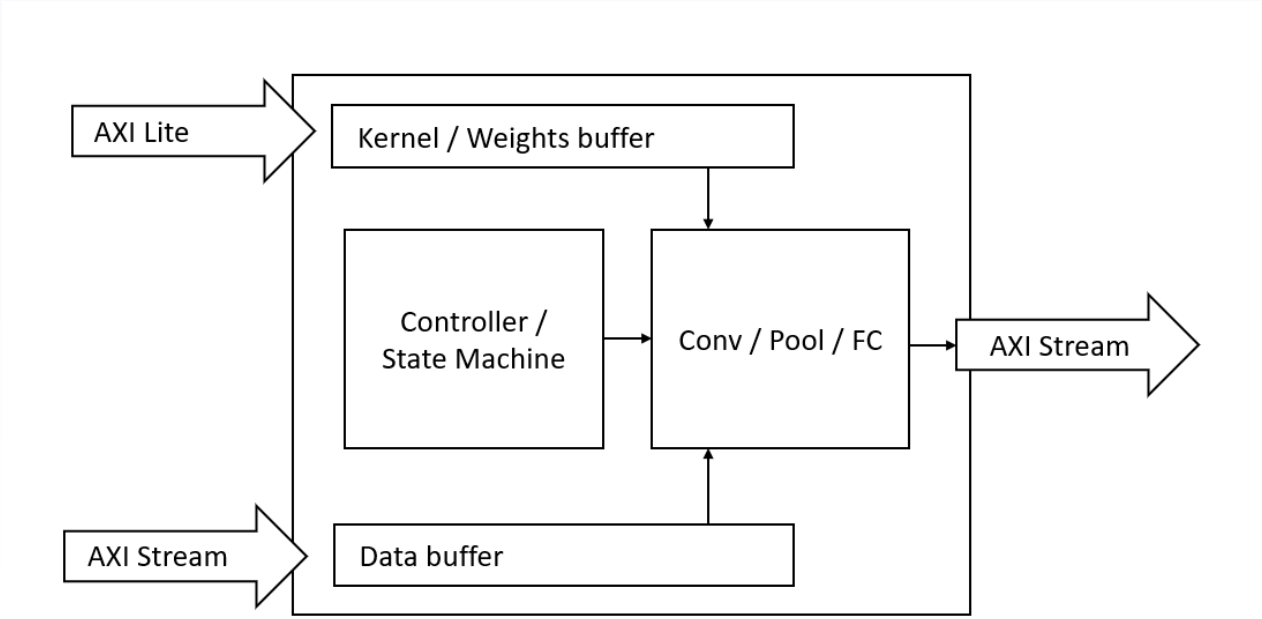


Figure: 3x3 neighborhood extractor architecture

Design – Processing Element

- **Single CNN Layer computation**
  - Convolution
  - Pooling (Max/Average)
  - Fully Connected
- **Stream input/output (AXI Stream)**
  - Data enters block
  - Calculation preformed
  - Data streamed deeper into network
- **Memory Mapped input (AXI Lite)**
  - Pass configuration / control data
  - Kernels / weights



Current project development status

- **Top level data-path**
  - SW – HW communication
  - Input / Output infrastructure
  - Will be used as base for CNN data path

- **Taste of acceleration**
  - Simple Test Block
  - Gain over 10K integer array
  - PL uses 25% less clock cycles compared to PS
  - That's with no parallelism yet

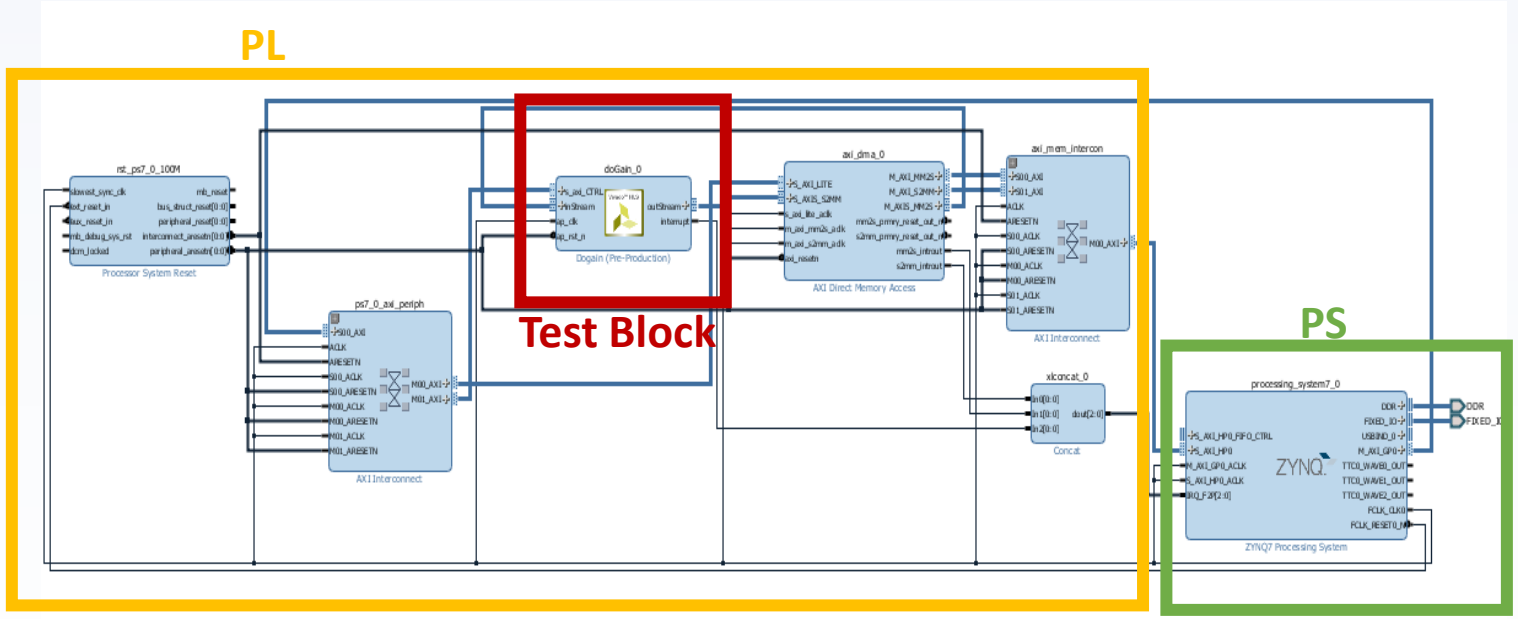


Figure: current project state. Vivado capture

```
Hello World
initializing doGain
initializing AxiDMA
Sending data to IP core slave
OutputPL took 107362 clock cycles.
OutputPS took 141368 clock cycles.
```

Figure: Terminal output. TeraTerm capture

Updated Timeline

Milestone	Description	Due/completion Date	Notes
CNN Background review	What is CNN?	01.12.2020	completed
Learning development tools	Verilog, Vivado, HLS, Zedboard, AXI	01.12.2020	completed
Operation Characterization	Complete System design	15.12.2020	completed
Top level Data Path infrastructure	implement main data flow interface	10.01.2020	completed
Mid-term presentation		18.01.2021	today
PE block	Implementation and testing	01.03.2021	
Sliding Window block	Implementation and testing	01.04.2021	
CNN integration	Integration of all blocks of the CNN Core and Testing.	15.04.2021	
TOP level integration	Integrating CNN in TOP design	01.05.2021	
Drivers	SW side of CNN	15.05.2021	
Submitting the poster		20.05.2021	
Performance improvements	Both HW and SW	01.06.2021	
CNN Simulation		15.06.2021	
Project submission		20.06.2021	