מס' פרויקט:
20-1-1-2187

שם הפרויקט:
מימוש וסימולציה של מאיץ למערכות לומדות על רכיב FPGA

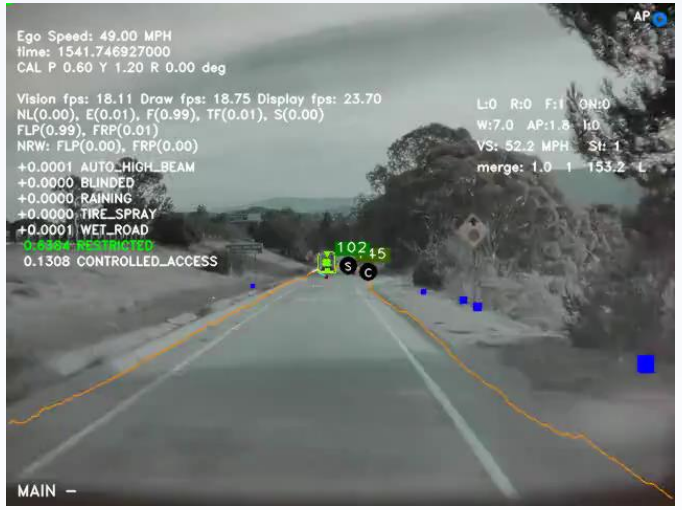
מגישים:
שי צבר 208723627
חיים גרודה 312562721

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 for autonomous cars, security systems, health and more.



Video: Tesla computer vision

CNN Structure

- **Convolution -** Feature extraction using filters
- **Activation –** Introduce nonlinearity
- **Pooling -** Reduce spatial dimensions
- **Fully Connected -** Final classification decision

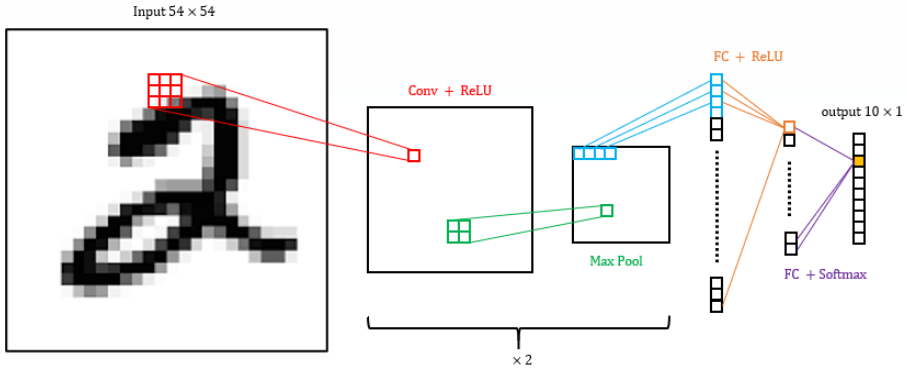


Figure: Implemented CNN model

Motivation and Objectives

Motivation

- High accuracy comes with high computation time. Approx. 90% of CNN computations are in the convolutional layers
- Computation resources cause high-power consumption
- Need to include such image recognition capabilities in embedded systems with tight real-time and power constraints

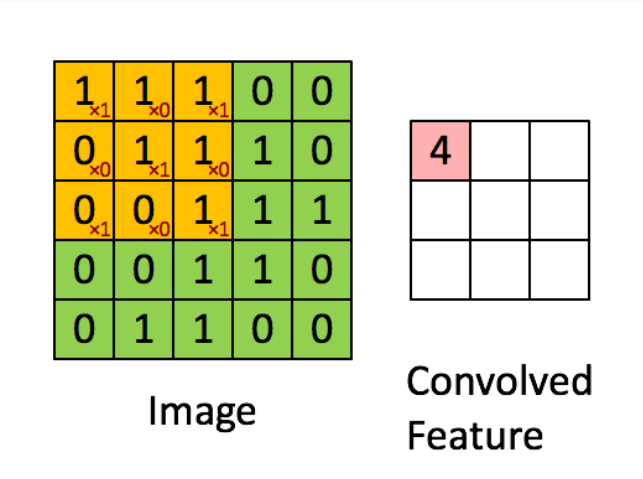


Figure: 2D convolution. [source]

Objectives

- **Acceleration** – by hardware parallelization
- **Accuracy** – reduce loss due to fixed point arithmetic
- **Power** – dedicated hardware processing elements

```
for(x=0;x<F;x++){
 for(y=0;y<E;y++){
  for(k=0;k<C;k++){
   for(i=0;i<R;i++){
    for(j=0;j<S;j++){
     for(m=0;m<M;m++){
      Output[m][x][y] +=
       Input[k][x+i][y+j] ×
       Weight[m][k][i][j]
} } } } } }
```

Figure: Nested for loop for convolution calculation

Method

ZedBoard FPGA

- Programmable logic
- Control design optimization
- Fast development

High Level Synthesis (HLS)

- RTL abstraction
- Focus on functionality
- Easily explore algorithmic changes
- Simple platform retargeting

Fixed Point Calculations:

- Better computation time
- Less hardware resources
- Possible loss of accuracy

Simulation:

- SW implementation of CNN
- Comparison time HW vs SW
- Compare accuracy HW vs SW
- Evaluate power consumption

Hardware Design

Processing system (PS):

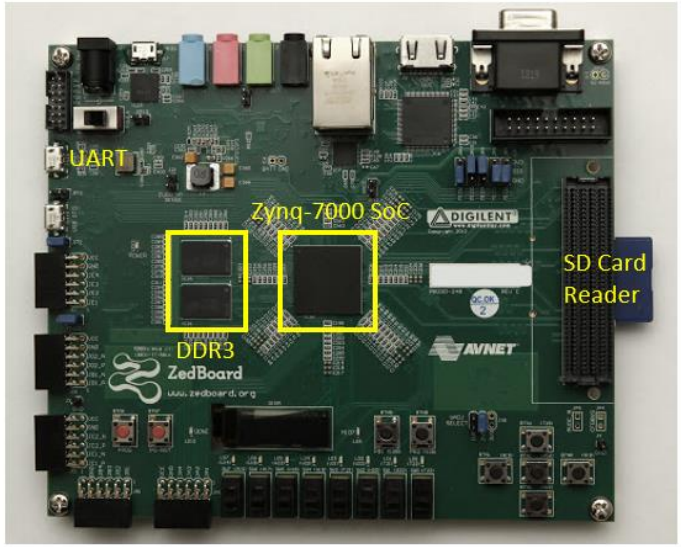
- ARM Cortex A9 667MHz CPU
- Input Output – SD card, UART
- Control and configuration
- DDR memory unit

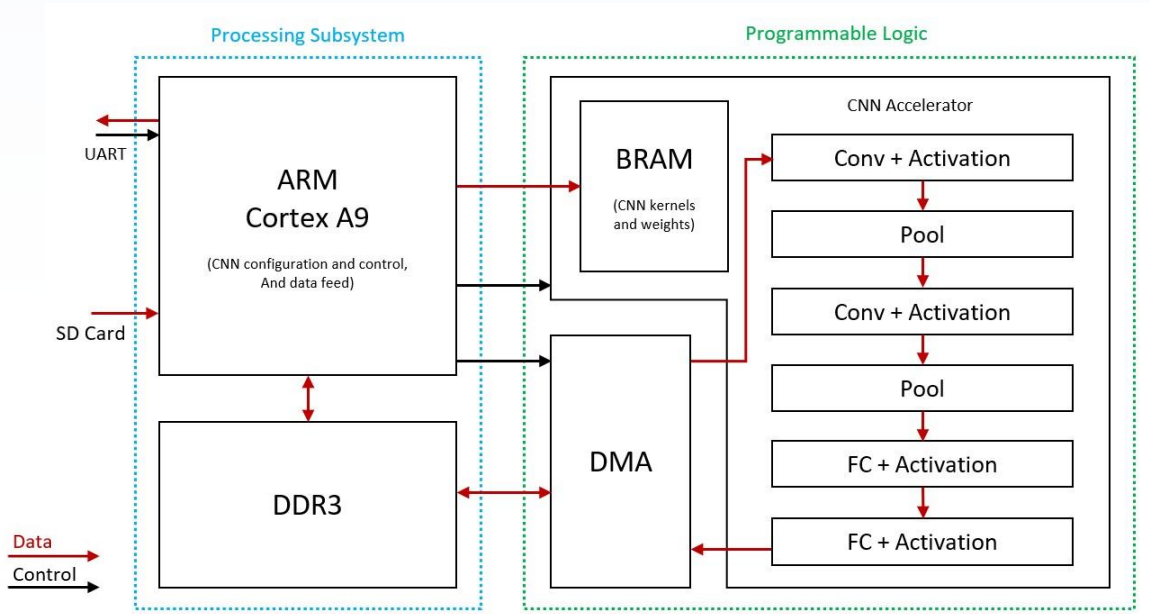
Programmable Logic (PL):

- Artix-7 FPGA with 6.6M logic cells
- DMA controller for data transfer
- Processing Element per CNN layer
- BRAM for kernels and weights

AXI interfaces

- Stream – high throughput (input data)
- Lite – static data (kernel/weights)

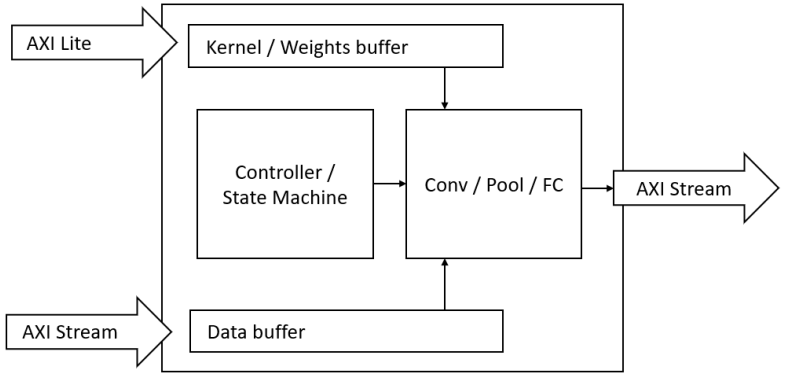




Hardware Design

Processing Element

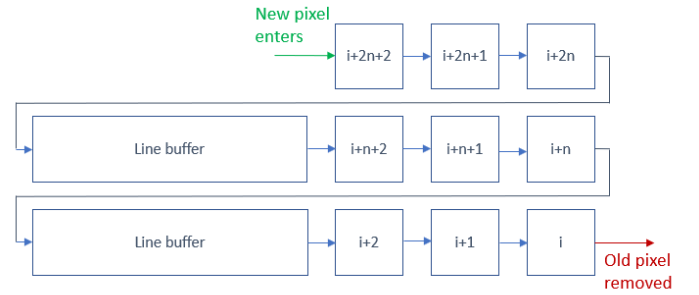
- Single CNN layer
- BRAM buffers
- FIFO implementation

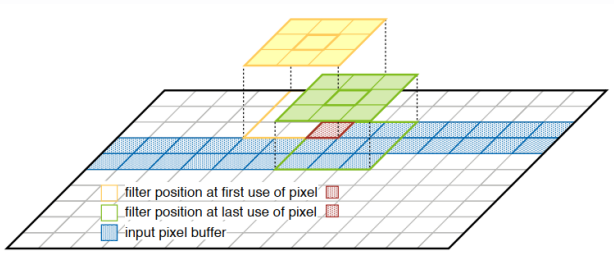


Figures: Processing Element design

Sliding Window

- Neighborhood extraction mechanism
- Memory utilization
- Data reuse – pixel transferred only once





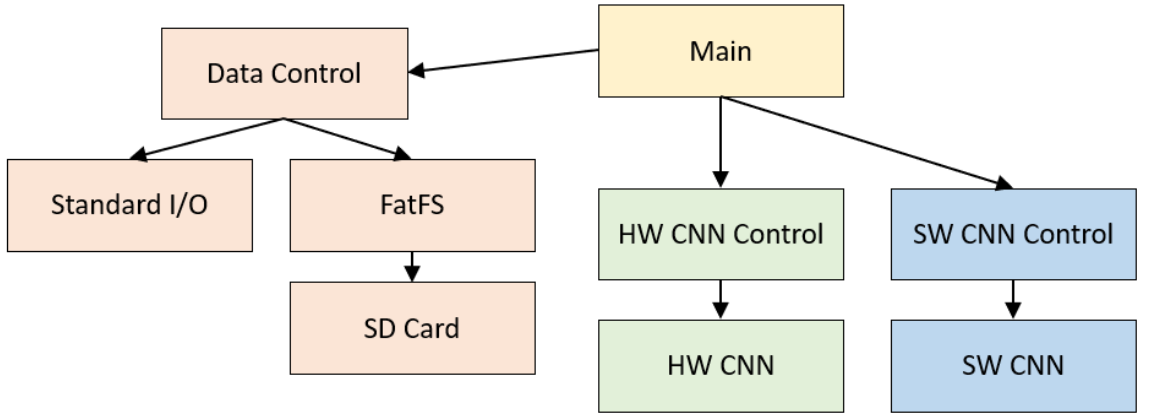
Figures: Data reuse / sliding window illustration

Software Design

- Cross platform: PC and FPGA
- Written in C in Vivado SDK

FPGA platform (ARM Cortex A9 CPU)

- Bare Metal (no OS)
- FatFS – Free file system software module
- Maximum Stack (64Kb)



PC platform (Intel i7 CPU)

- Linux based
- Single threaded
- Possible Simulation of HW calculations

HW Implementation

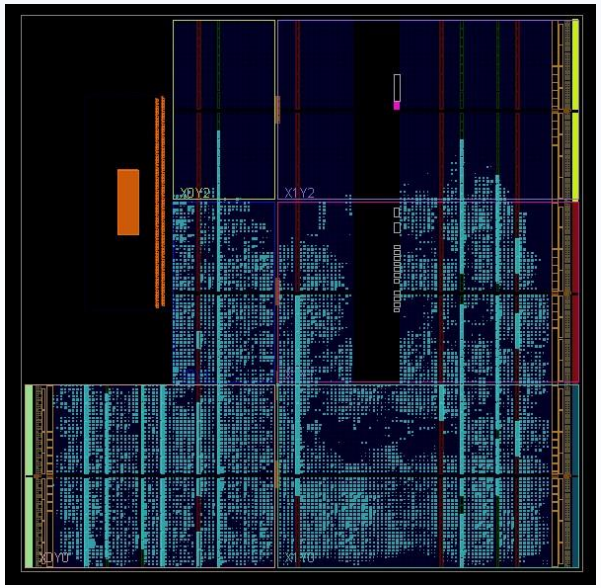


Figure: Implemented Design

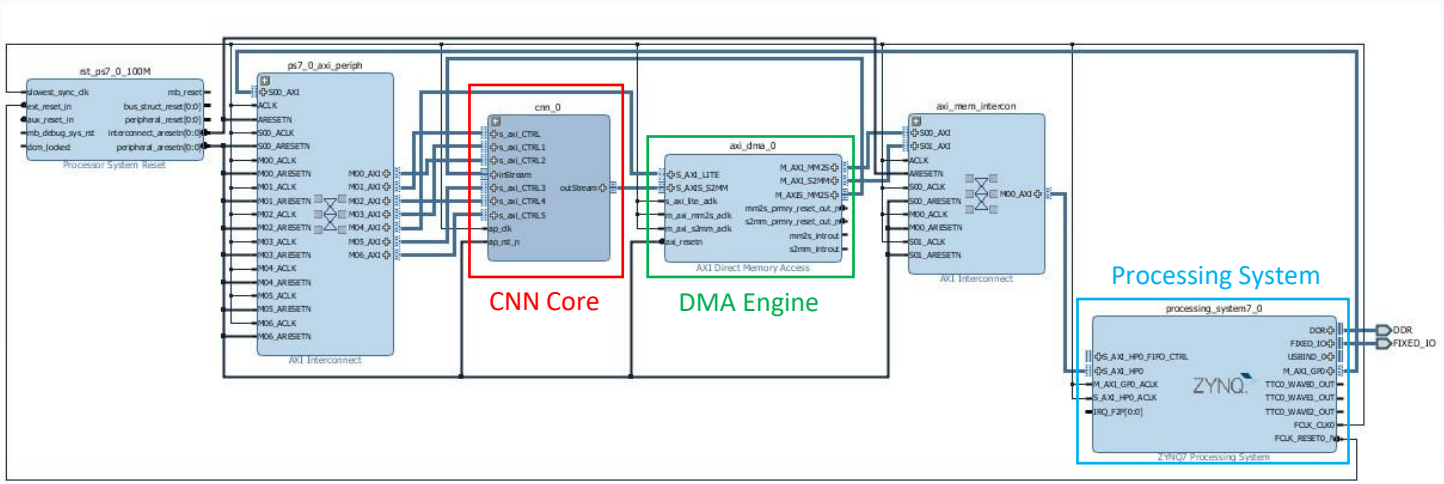


Figure: TOP block design

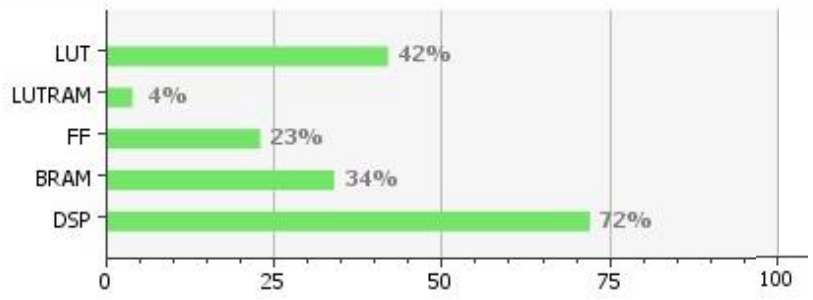


Table: FPGA utilization

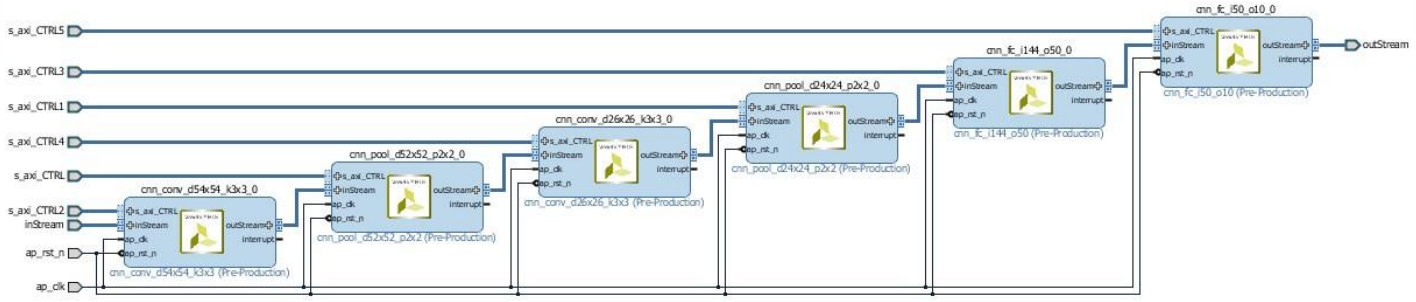


Figure: CNN core block design

Simulation Results

CNN Simulation

- Python Keras deep learning library
- MNIST dataset, 10K test images
- Kernels and weights fed to CNN

	Floating point (FP64)	Fixed point (Q32.10)
Accuracy	96.03%	96.02%
Certainty	98.83%	98.82%

Table: CNN accuracy using different arithmetic

Measurements

- Avg. image processing time
- Avg. accuracy
- Avg. certainty
- Vivado power estimation

	PS	PL
Dynamic Power	77%	23%
Total Power	71%	29%

Table: Vivado power estimation results

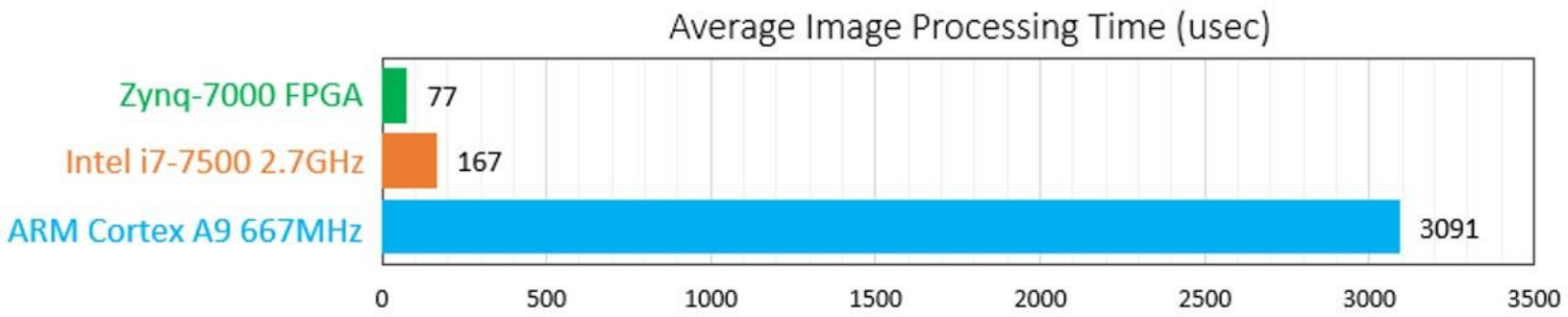


Table: Time comparison

Conclusions and Future Work

Conclusions

- Significant acceleration compared to SW
- No loss of accuracy with fixed point
- Significant power conservation

Future Work

- Add convolution channels (RGB)
- Compare with multi threaded SW implementations
- Test accuracy loss for different fixed-point schemes

Appendix and References

- Project repository on GitHub
- ZynqNet - An FPGA-Accelerated Embedded CNN
- Science Direct - Convolutional Neural Networks