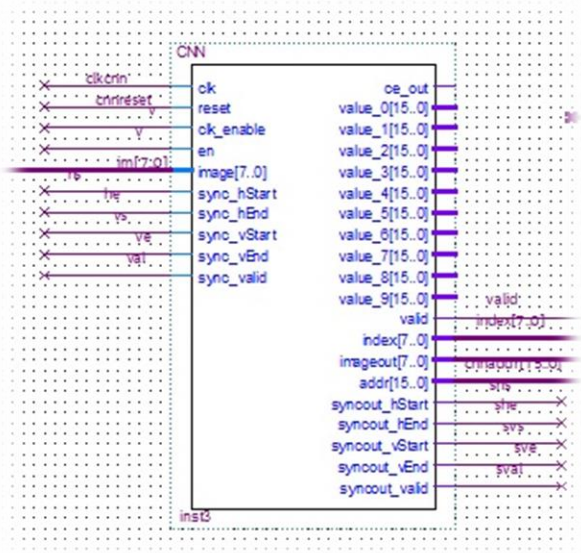
экспонента

Центр Инженерных Технологий
и Моделирования

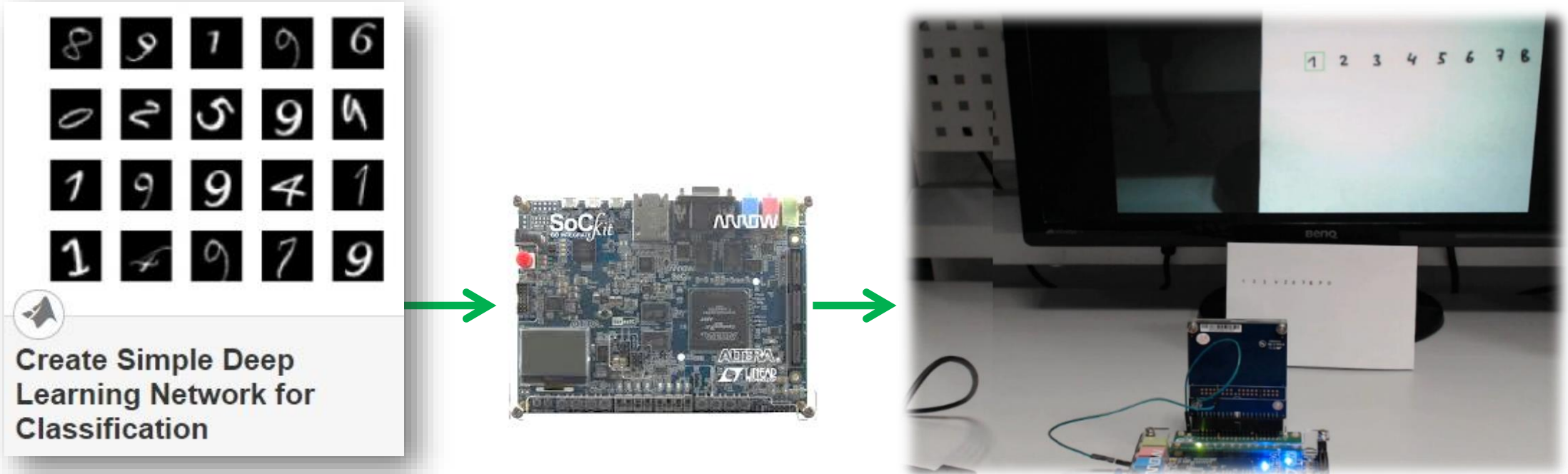


АППАРАТНАЯ РЕАЛИЗАЦИЯ СВЕРТОЧНОЙ НЕЙРОСЕТИ НА ПЛИС ИСПОЛЬЗУЯ МОДЕЛЬНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ

Александр Воробьев
Инженер ЦИТМ экспонента

Реализация сверточной нейронной сети ПЛИС

- **Цели проекта:**
  - Запуск нейронной сети для распознавания цифр на ПЛИС
  - Разработка рабочего процесса реализации сети на ПЛИС



Модельно-Ориентированное Проектирование

Основа – MATLAB / Simulink

Модельно-ориентированное проектирование

- **Верификация** системы на более высоких уровнях абстракции
- **Быстрый анализ** альтернативных вариантов реализации алгоритмов
- Описание проекта не привязано к аппаратной реализации

Среда разработки – MATLAB и Simulink

- Наличие обширных **библиотек,** подключение внешнего кода
- Возможность **генерация кода** C, C++, CUDA и HDL-кода
- **Нет разрыва** между алгоритмическим описанием и HDL-описанием системы
- **Сокращение времени разработки** по сравнению с традиционными подходами

Процесс разработки согласно МОП

- **Этапы разработки согласно концепции Модельно ориентированного проектирования**

1) Создание модели, разработка алгоритма
2) Подготовка к аппаратной реализации
3) Автоматическая генерация кода
4) Подключение кода в САПР для ПЛИС

Этапы разработки при модельно-ориентированном проектировании

Проектирование архитектуры нейронной сети

- **Этап 1. Проектирование архитектуры нейронной сети**

Deep Learning Toolbox

1) Исходная сеть: **15 слоев** – точность **99.8**%

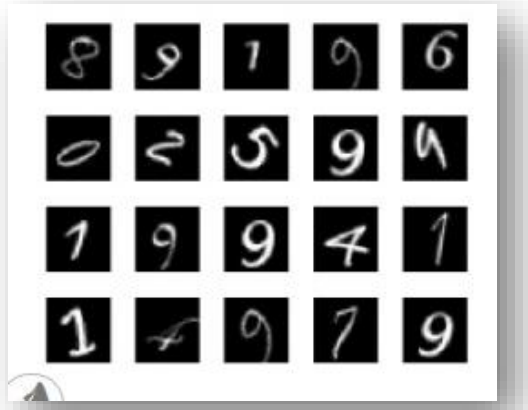
2) Оптимизация архитектуры сети > 2 раза: **7 слоев – 98.8**%

```
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,32,'Padding','same')
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];
```

Оптимизированная архитектура сети

```
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(5,20)
    reluLayer()
    maxPooling2dLayer(2,'Stride'
    fullyConnectedLayer(10)
    softmaxLayer()
    classificationLayer()];
```

Пример цифр для классификации



Подготовка к аппаратной реализации

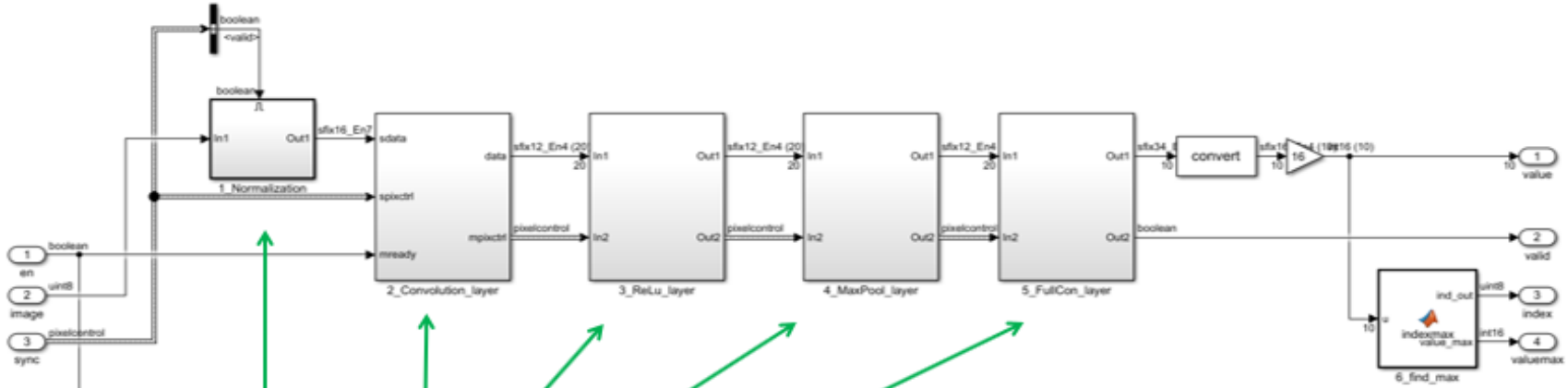
- **Этап 2. Подготовка к аппаратной реализации**

Simulink

1) Перевод алгоритма для работы с потоковыми данными

2) Перевод алгоритма в фиксированную точку – 16 бит

По пиксельная развертка изображения



1	''	Image Input	28x28x1 images with 'zerocenter' normalization
2	''	Convolution	20 5x5 convolutions with stride [1 1] and padding
3	''	ReLU	ReLU
4	''	Max Pooling	2x2 max pooling with stride [2 2] and padding [0
5	''	Fully Connected	10 fully connected layer
6	''	Softmax	softmax <- от произвольных значений к вероятностям
7	''	Classification Output	crossentropyex <- метка для максимального значения

Модель алгоритма нейронной сети в Simulink

Автоматическая генерация кода

- **Этап 3. Автоматическая генерация кода**

HDL Coder

1) Автоматическая генерация кода

2) Оптимизация используемых ресурсов ПЛИС

Altera SoCKit

Аппаратная платформа:

- Altera SoCKit Development Kit с **Cyclone V** SoC (5CSXFC6D6F31)
- Количество логических ячеек: 110К LE (**41,9К ALM**)
- **Количество умножителей** - **112**
- Объем встроенной памяти - **5,140 Кбит**

Камера D8M-GPIO

Автоматическая генерация кода

Этап 3.1 Автоматическая генерация кода

- Слой «Convolution_layer» – **500** умножителей (**112** доступно аппаратно)

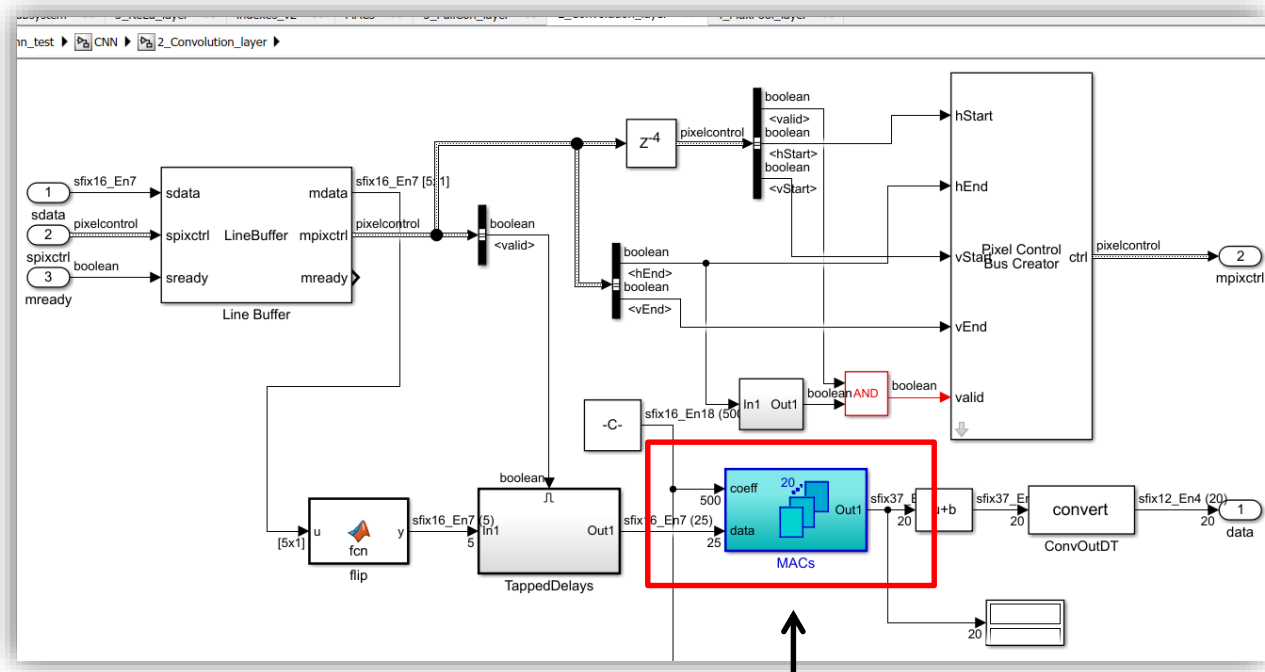
```
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(5,20)
    reluLayer()
    maxPooling2dLayer(2,'Stride
```

5x5x20 = 500

Используемые ресурсы ПЛИС - без оптимизации

Multipliers	600
Adders/Subtractors	1038
Registers	29557
Total 1-Bit Registers	480163
RAMs	109
Multiplexers	1107
I/O Bits	232
Static Shift operators	0
Dynamic Shift operators	0

2_Convolution_layer



Умножение с накоплением

Оптимизация ресурсов

- **Этап 3.2 Оптимизация используемых ресурсов ПЛИС**

- SharingFactor = 20

- Слой «Convolution_layer» – **25 умножителей**

Без оптимизации

Multipliers	600
Adders/Subtractors	1038
Registers	29557
Total 1-Bit Registers	480163
RAMs	109
Multiplexers	1107
I/O Bits	232
Static Shift operators	0
Dynamic Shift operators	0

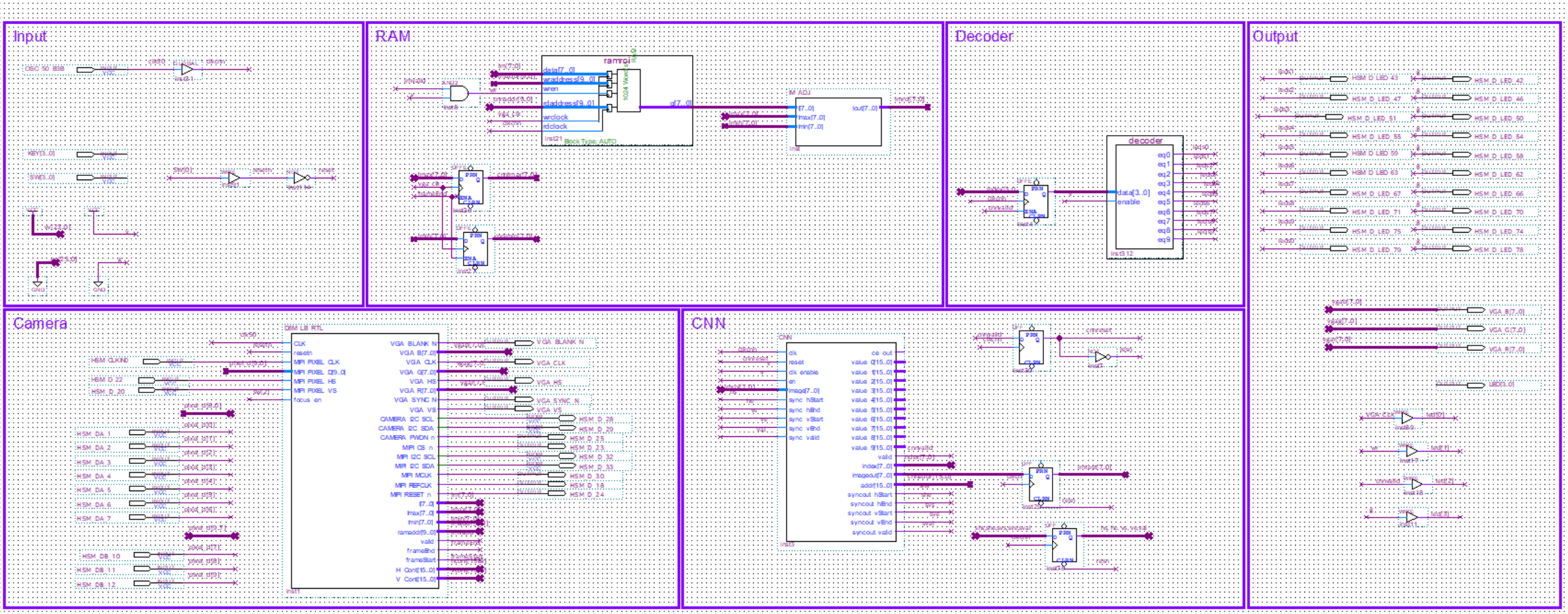
Совместное использование ресурсов

Multipliers	125	4.8 раза
Adders/Subtractors	613	
Registers	4306	6.8 раза
Total 1-Bit Registers	77343	6.2 раза
RAMs	133	
Multiplexers	1210	
I/O Bits	232	
Static Shift operators	0	
Dynamic Shift operators	0	

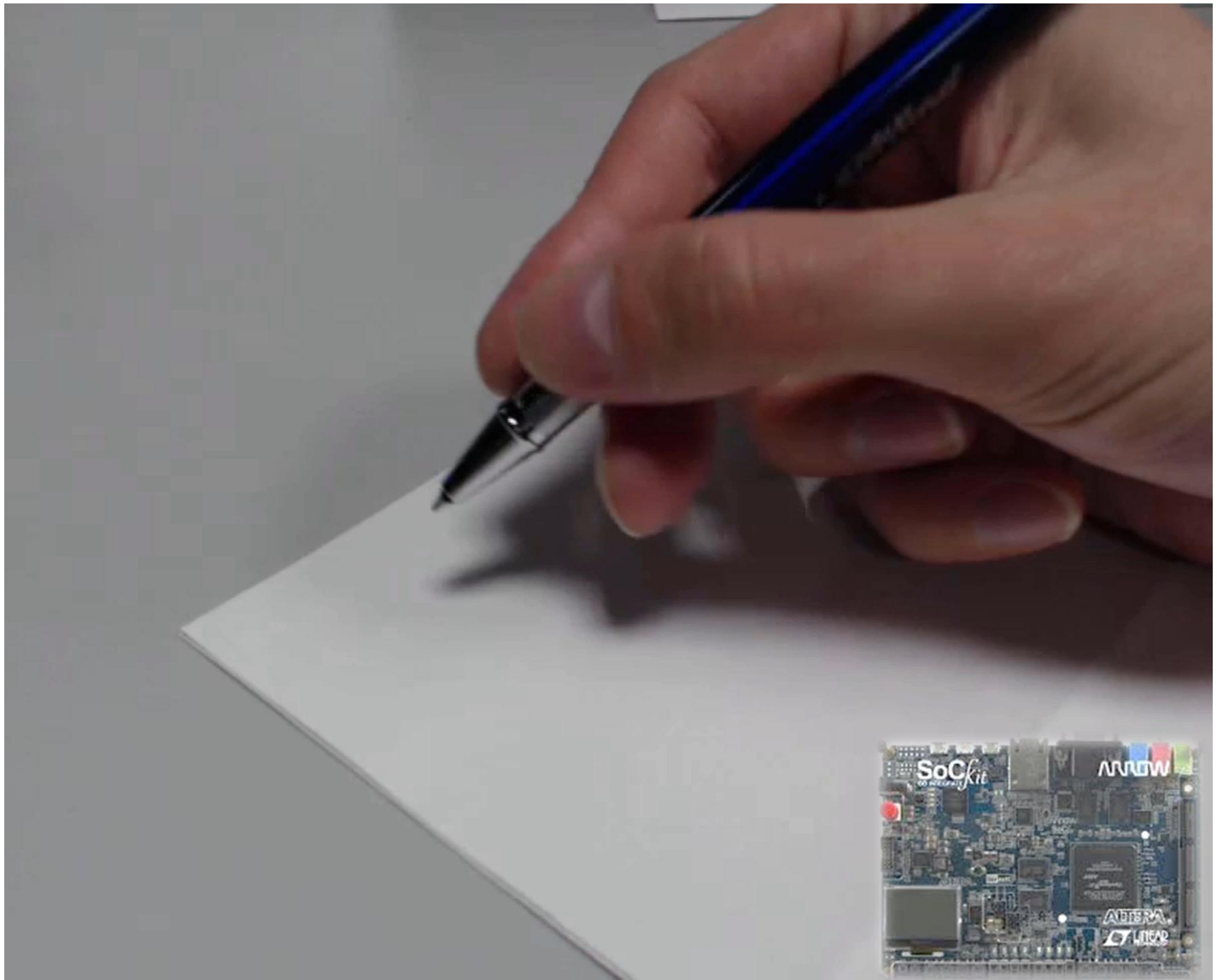
Отчет HDL Coder- используемые ресурсы ПЛИС

Подключение модуля сети в Quartus

- **Этап 4 Подключен кода в САПР для ПЛИС**



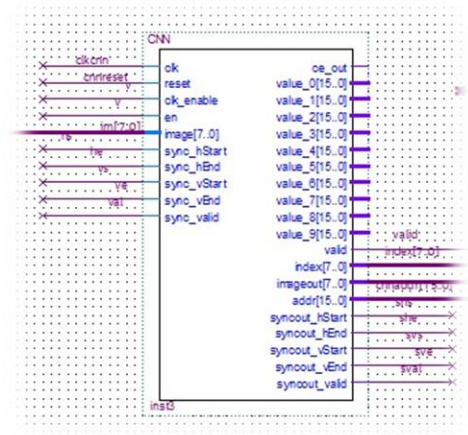
Тестирование алгоритма сети на отладочной плате



Используемые ресурсы ПЛИС

Flow Summary	
<<Filter>>	
Flow Status	Successful - Wed Jun 20 16:18:53 2018
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	SocKit_golden_top
Top-level Entity Name	SoCKit_TOP_CAM_CNN_TEST
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	30,552 / 41,910 (73 %)
Total registers	18357
Total pins	84 / 499 (17 %)
Total virtual pins	0
Total block memory bits	870,019 / 5,662,720 (15 %)
Total DSP Blocks	112 / 112 (100 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	1 / 15 (7 %)
Total DLLs	0 / 4 (0 %)

Результаты проекта



- **Цели проекта:**
  - Запуск нейронной сети для распознавания цифр на ПЛИС
  - Разработка рабочего процесса реализации сети на ПЛИС
- **Полученные результаты:**
  - **Реализован алгоритм** нейронной сети на ПЛИС
  - **Отработан процесс** по реализации сети на ПЛИС на базе МОП
  - Найдена оптимальная архитектура сети по соотношению точность/сложность
  - Выработана методика оптимизации HDL-кода для сверхточных сетей
- **Дальнейшие развитие проекта:**
  - Обучить сеть для распознавание дорожных знаков
  - Распознавание автомобильных номеров

ЦИТМ «Экспонента»

- https://exponenta.ru https://matlab.ru

  Экспертиза по:

  - Обработке изображений и компьютерному зрению
  - Нейронным сетям и машинному обучению
  - Системам управления
  - Цифровой обработке сигналов и система связи
  - По развертыванию алгоритмов на встраиваемые системах