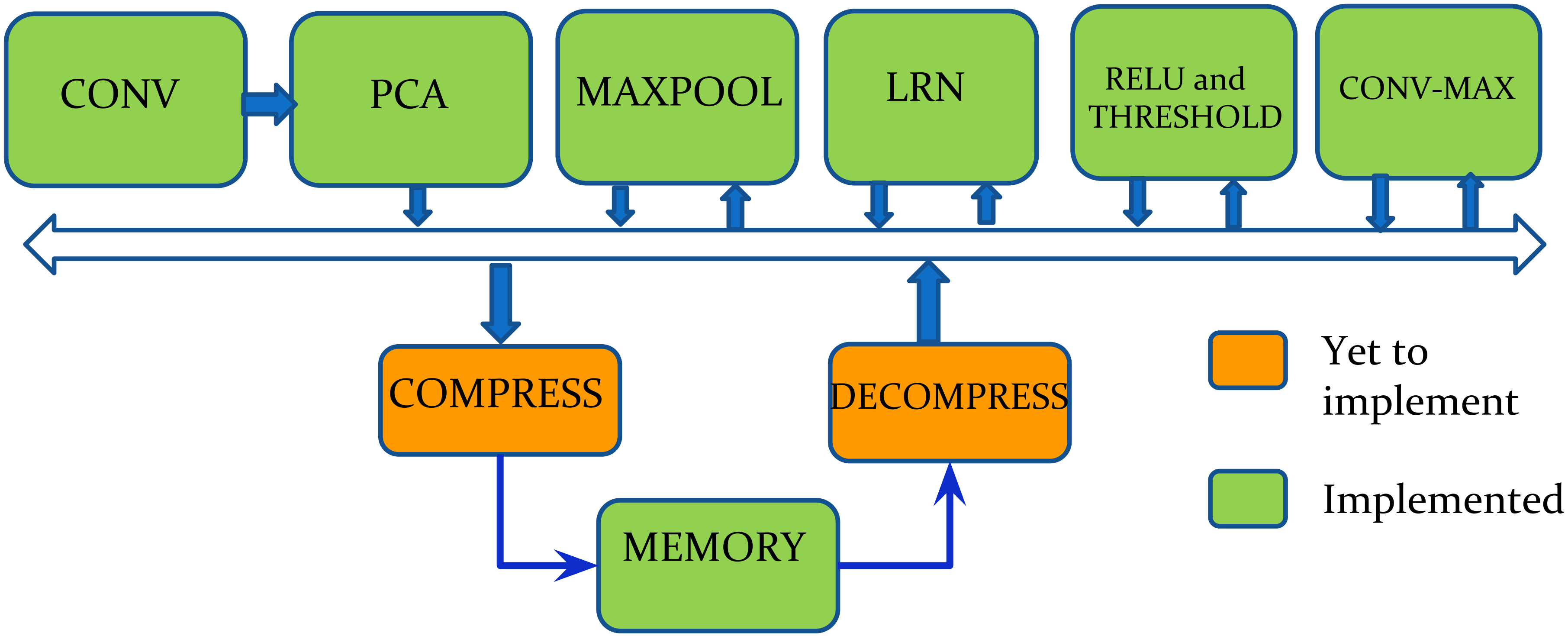
Convolutional Neural Network on FPGA

by:
Arjun Rakesh Gandhi
Mahmoud Khallaf-Allah
Venkatesh Mahadevan
Roberto Dicecco

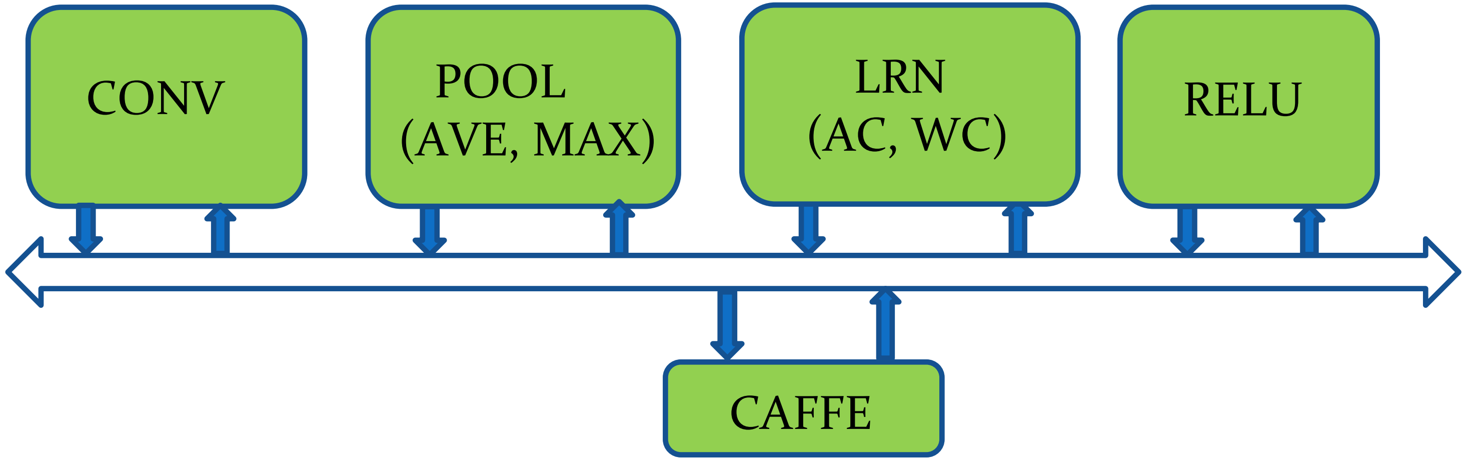
System-Level Architecture (initial)



🟧 Yet to implement

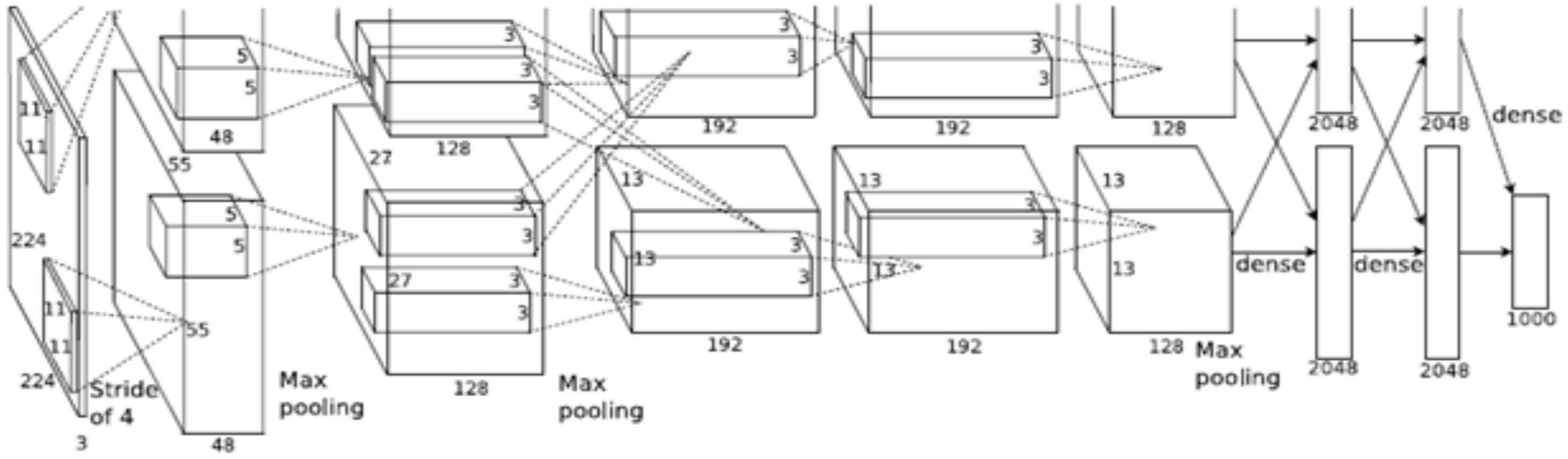
🟩 Implemented

System-Level Architecture (final)



🟩 Implemented

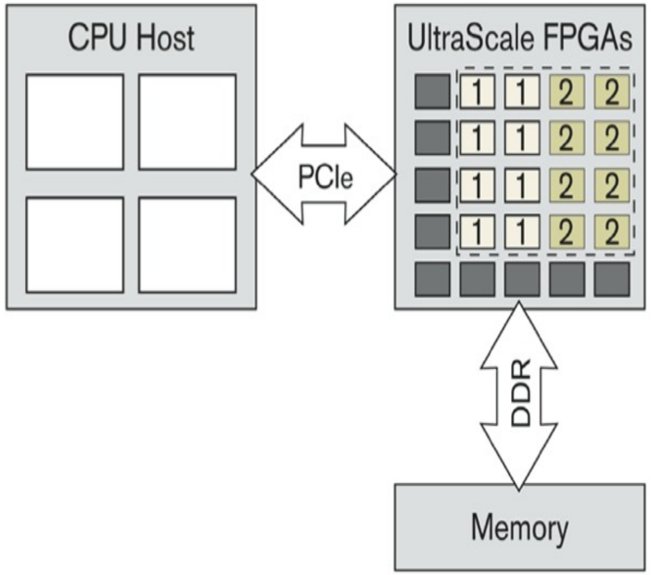
AlexNet Implementation using Caffe layers on FPGA

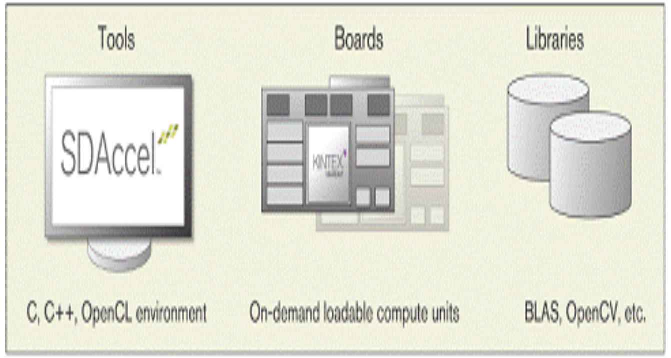


- Implementing CONV, POOL, LRN, RELU
- 5 CONV, 3 POOL, 2 LRN, 7 RELU (in hardware)
- Rest of the layers use existing CPU implementation

Moving to SDAccel

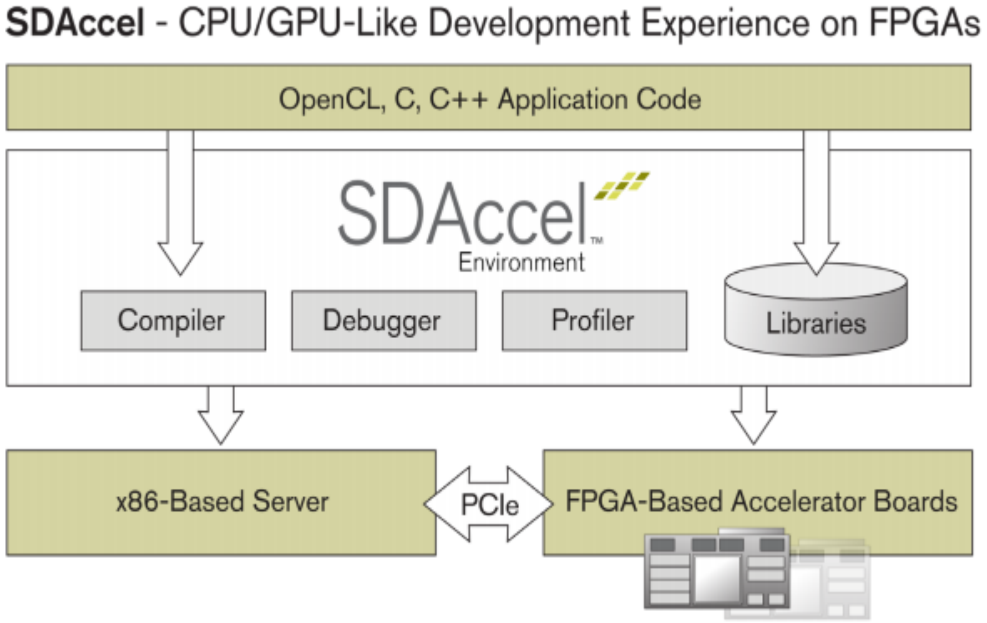
- **Why SDAccel?**
  - Provides CPU-GPU like platform to increase performance.
  - Use OpenCl APIs to increase parallelism.
  - Burden of context switching is on OpenCl.
  - Facilitate integration with Caffee library.
- **Re-implementing four main layers**





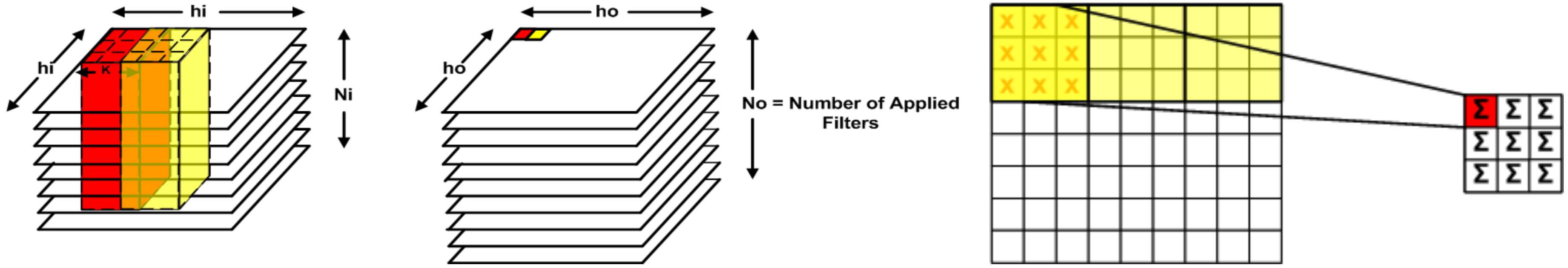
SDAccel execution flow

- Host Code:
  - Connect to platform and compute device.
  - Create a compute context.
  - Create the compute kernel and Allocate memory.
  - Execute Kernel.
  - Read results.


SDAccel - CPU/GPU-Like Development Experience on FPGAs
OpenCL, C, C++ Application Code
SDAccel Environment
Compiler
Debugger
Profiler
Libraries
x86-Based Server
PCIe
FPGA-Based Accelerator Boards

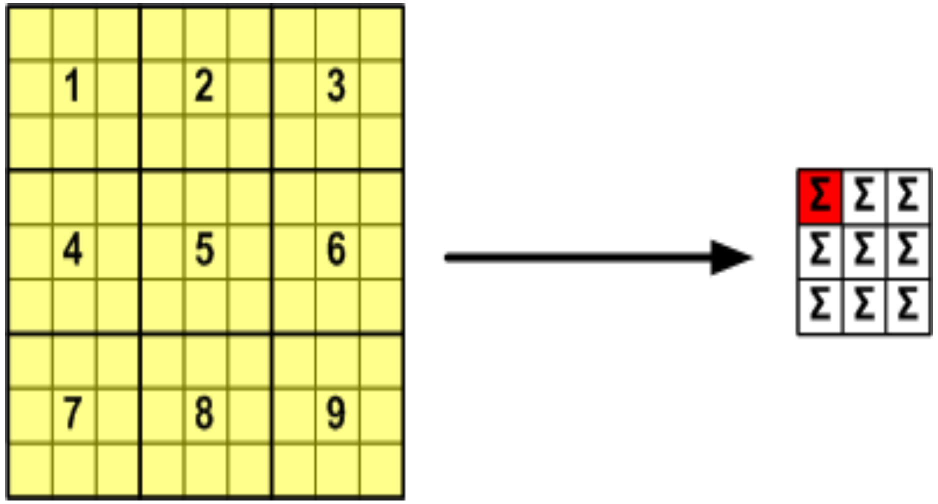
- Kernel Code
  - Execute Kernel function
  - Write output back

Convolutional Layer



- Element-wise multiplication between filter elements and input data.
- Filter is applied as a 3-D kernel.
- The final result will be the average of all summed outputs.
- Output Size is: $ho = \frac{hi - k + 2 * padding}{Stride} + 1$

Convolutional Layer (Cont.)



- Function input: **Data input , Filter weights**.
- Function Output: **Data output.**
- Configurations: **Input size, Filter size, Stride.**
- Two outer loops, and Two inner loops.
- Pipelining directives are used.

Convolutional Layer (Cont.)

Kernel Name	Target Freq	Estimated Freq	Start Interval	Best Case	Avg Case	Worst Case	FF	LUT	DSP	BRAM
Conv1	167 MHZ	144 MHZ	239837	239836	239836	239836	10652	10614	12	138
Conv2	167 MHZ	178 MHZ	11350	11349	11349	11349	4239	3588	11	4
Conv3	167 MHZ	178 MHZ	1329	1328	1328	1328	2978	2480	11	3

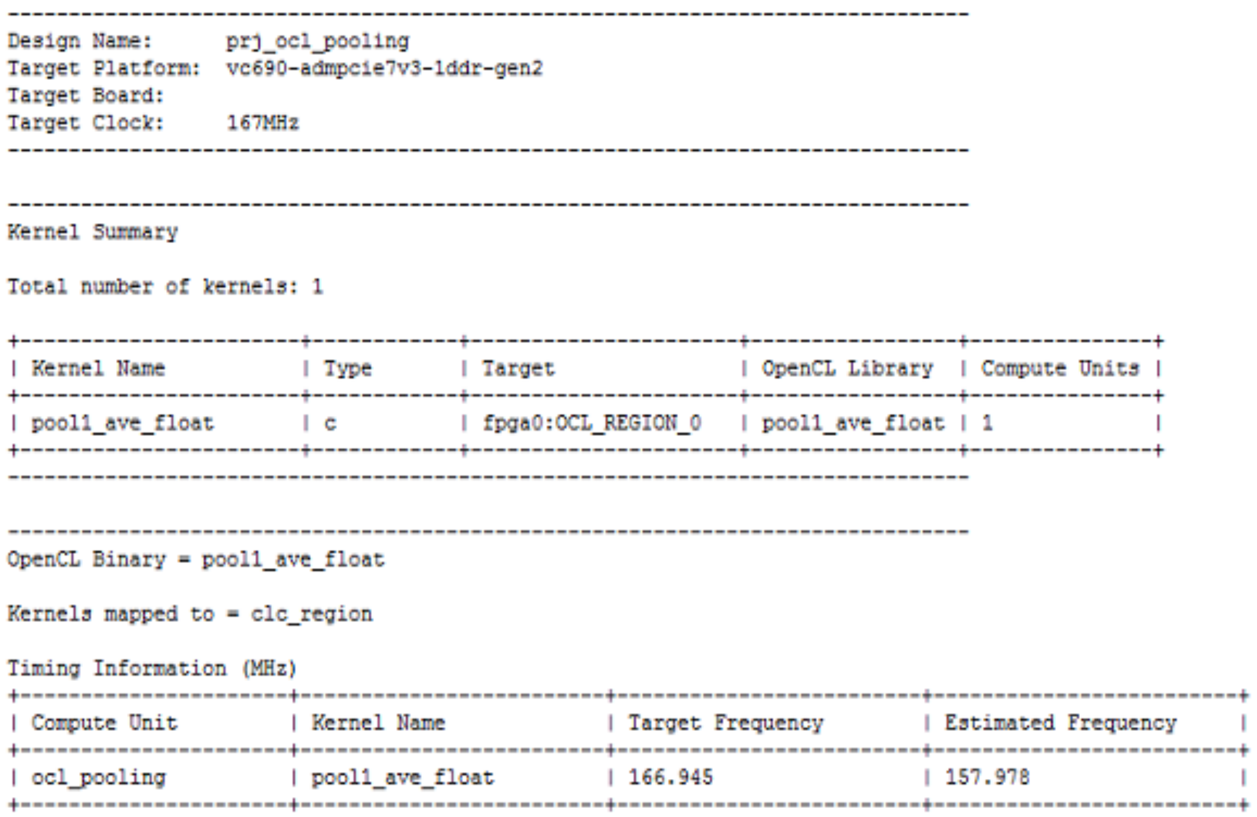
Pooling Layer

- Given an input image with width IWIDTH and height IHEIGHT, and a mask of width MASK_WIDTH and height MASK_HEIGHT, the pooling layer divides the input image into sections of the same size as the mask size.
- It then computes either the maximum or the average of the values in the region of the input encompassed by the mask.
- The final output has width OWIDTH and height OHEIGHT.

Pooling layer (contd ...)

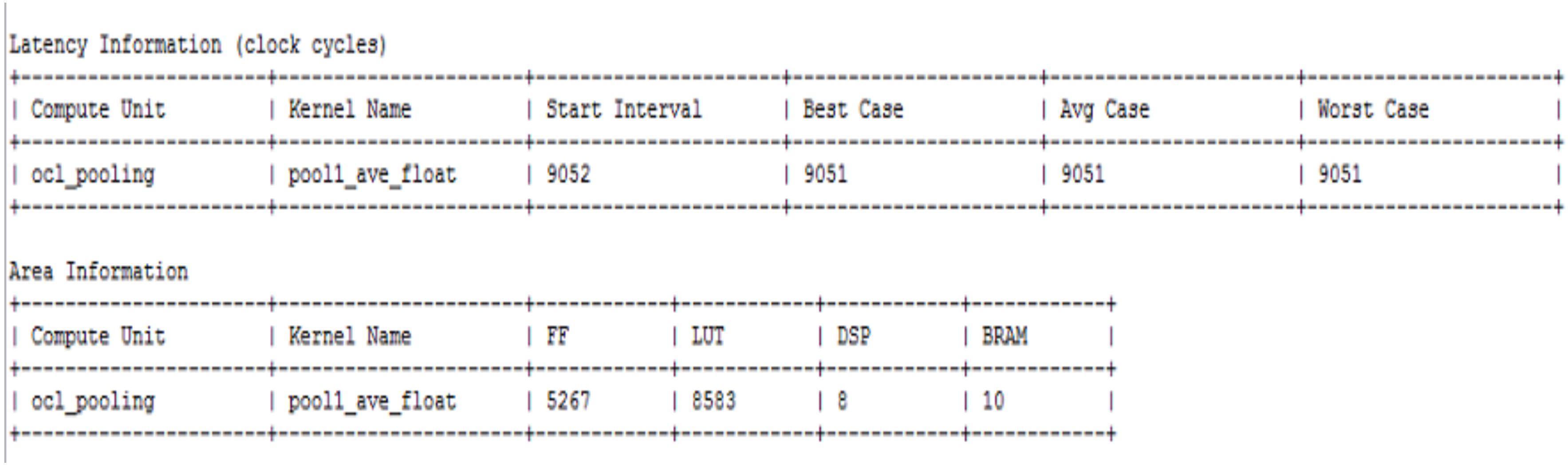
- As per Alex net model, a Pooling layer with max size of 55 x 55 is required.
- For Alex net’s requirement, the Pooling function can be either the max function or the average function.
- 3 instances used in Alex net implementation.
- Pipelining is used for increasing throughput and reducing latency.

Pooling Resource utilization and Timing information (ave)



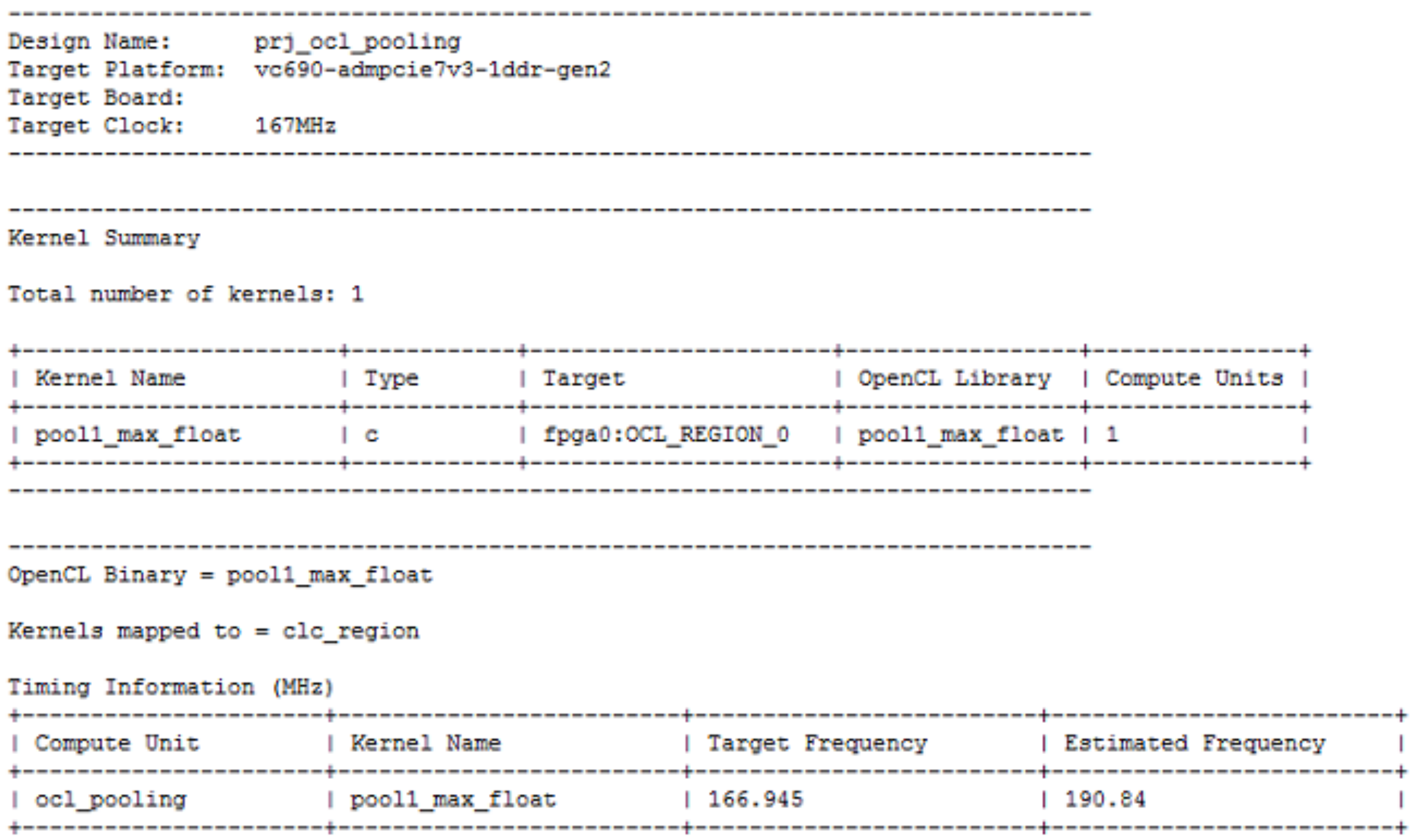
```
-------------------------------------------------------------------------------
Design Name:      prj_ocl_pooling
Target Platform:  vc690-admpcie7v3-1ddr-gen2
Target Board:
Target Clock:     167MHz
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
Kernel Summary

Total number of kernels: 1

+-----------------------+------------+----------------------+-----------------+---------------+
| Kernel Name           | Type       | Target               | OpenCL Library  | Compute Units |
+-----------------------+------------+----------------------+-----------------+---------------+
| pool1_ave_float       | c          | fpga0:OCL_REGION_0   | pool1_ave_float | 1             |
+-----------------------+------------+----------------------+-----------------+---------------+
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
OpenCL Binary = pool1_ave_float

Kernels mapped to = clc_region

Timing Information (MHz)
+----------------------+--------------------------+--------------------------+--------------------------+
| Compute Unit         | Kernel Name              | Target Frequency         | Estimated Frequency      |
+----------------------+--------------------------+--------------------------+--------------------------+
| ocl_pooling          | pool1_ave_float          | 166.945                  | 157.978                  |
+----------------------+--------------------------+--------------------------+--------------------------+
```

Pooling Latency and Area information (ave)



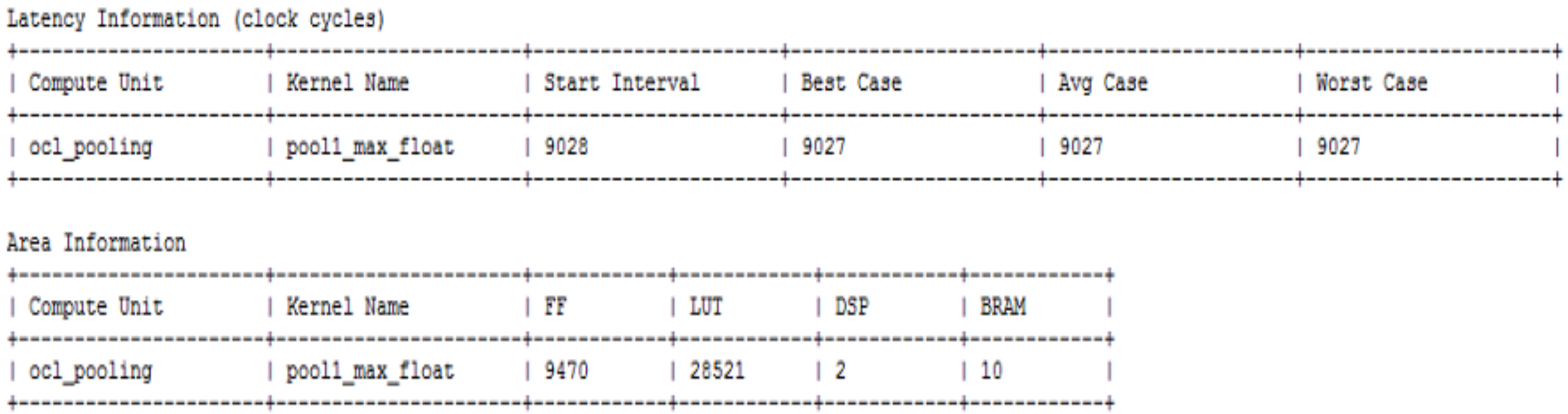
```
Latency Information (clock cycles)
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| Compute Unit        | Kernel Name         | Start Interval      | Best Case           | Avg Case            | Worst Case          |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| ocl_pooling         | pool1_ave_float     | 9052                | 9051                | 9051                | 9051                |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+

Area Information
+---------------------+---------------------+-----------+-----------+-----------+-----------+
| Compute Unit        | Kernel Name         | FF        | LUT       | DSP       | BRAM      |
+---------------------+---------------------+-----------+-----------+-----------+-----------+
| ocl_pooling         | pool1_ave_float     | 5267      | 8583      | 8         | 10        |
+---------------------+---------------------+-----------+-----------+-----------+-----------+
```

Pooling Resource utilization and Timing information (max)



```
-------------------------------------------------------------------------------
Design Name:      prj_ocl_pooling
Target Platform:  vc690-admpcie7v3-1ddr-gen2
Target Board:
Target Clock:     167MHz
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
Kernel Summary

Total number of kernels: 1

+-----------------------+------------+----------------------+-----------------+---------------+
| Kernel Name           | Type       | Target               | OpenCL Library  | Compute Units |
+-----------------------+------------+----------------------+-----------------+---------------+
| pool1_max_float       | c          | fpga0:OCL_REGION_0   | pool1_max_float | 1             |
+-----------------------+------------+----------------------+-----------------+---------------+
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
OpenCL Binary = pool1_max_float

Kernels mapped to = clc_region

Timing Information (MHz)
+----------------------+------------------------+------------------------+------------------------+
| Compute Unit         | Kernel Name            | Target Frequency       | Estimated Frequency    |
+----------------------+------------------------+------------------------+------------------------+
| ocl_pooling          | pool1_max_float        | 166.945                | 190.84                 |
+----------------------+------------------------+------------------------+------------------------+
```

Pooling Latency and Area information (max)



```
Latency Information (clock cycles)
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
| Compute Unit       | Kernel Name        | Start Interval     | Best Case          | Avg Case           | Worst Case         |
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
| ocl_pooling        | pool1_max_float    | 9028               | 9027               | 9027               | 9027               |
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+

Area Information
+--------------------+--------------------+----------+----------+----------+----------+
| Compute Unit       | Kernel Name        | FF       | LUT      | DSP      | BRAM     |
+--------------------+--------------------+----------+----------+----------+----------+
| ocl_pooling        | pool1_max_float    | 9470     | 28521    | 2        | 10       |
+--------------------+--------------------+----------+----------+----------+----------+
```

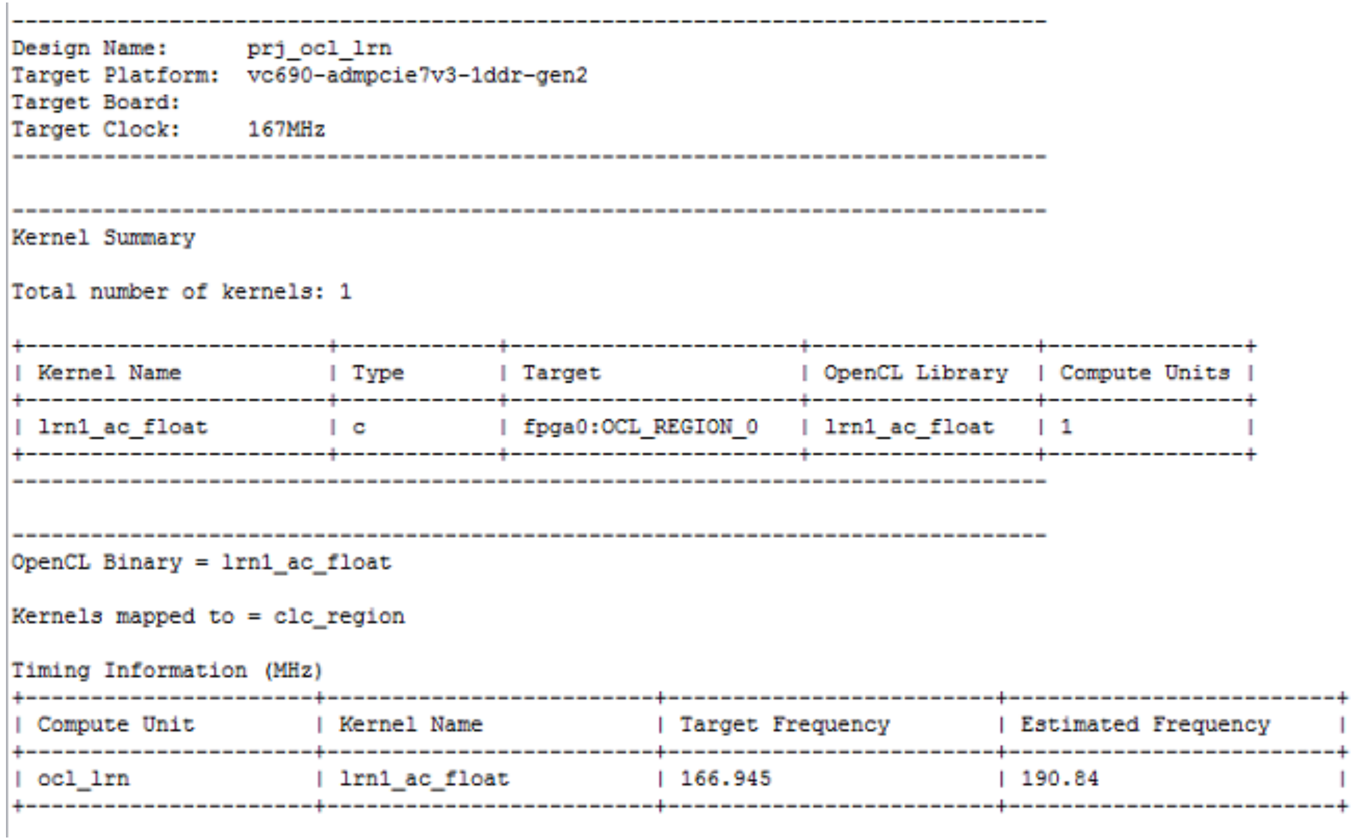
LRN Layer

- Given an input image with width IWIDTH and height IHEIGHT, and a mask of width MASK_WIDTH and height MASK_HEIGHT, the LRN layer divides the input image into sections of the same size as the mask size.
- The LRN layer performs neural activity inhibition by squaring all the values in the mask region,multiplying them by alpha, and dividing them by the mask size to create a scaling factor.
- This scaling factor is then raised to beta, and applied to each value in the mask region.

LRN layer (contd ...)

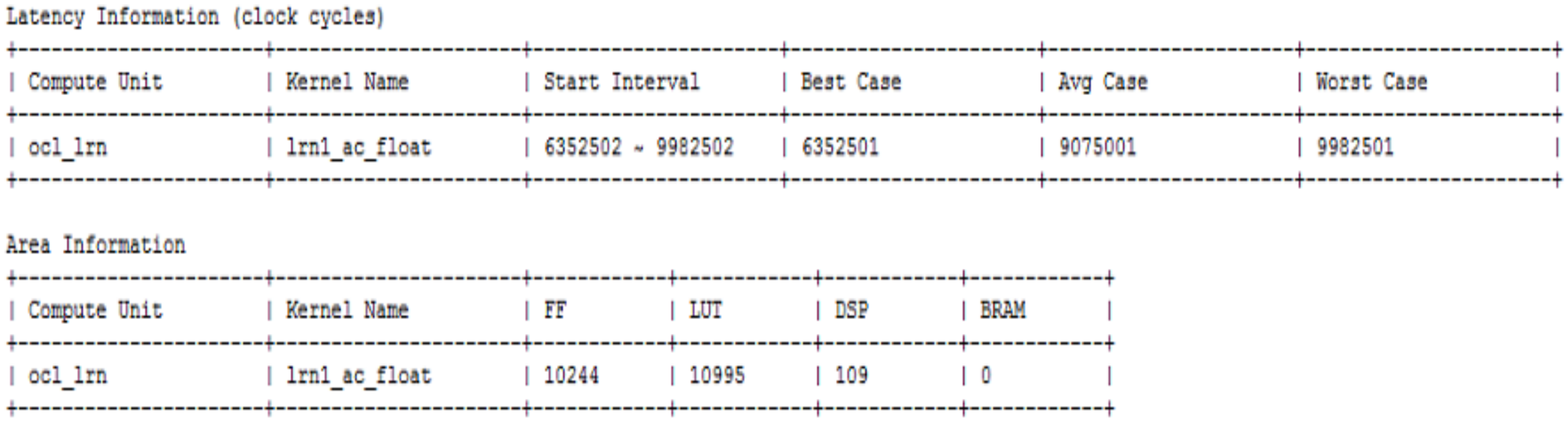
- As per Alex net model, an LRN layer with max size of 10x256x27x27 is required.
- For Alex net’s requirement, the LRN layer can be either a cross-channel or a within-channel implementation
- 2 instances used in Alex net implementation.
- Pipelining is used for increasing throughput/ reducing overall latency.

LRN Resource utilization and Timing information (AC)



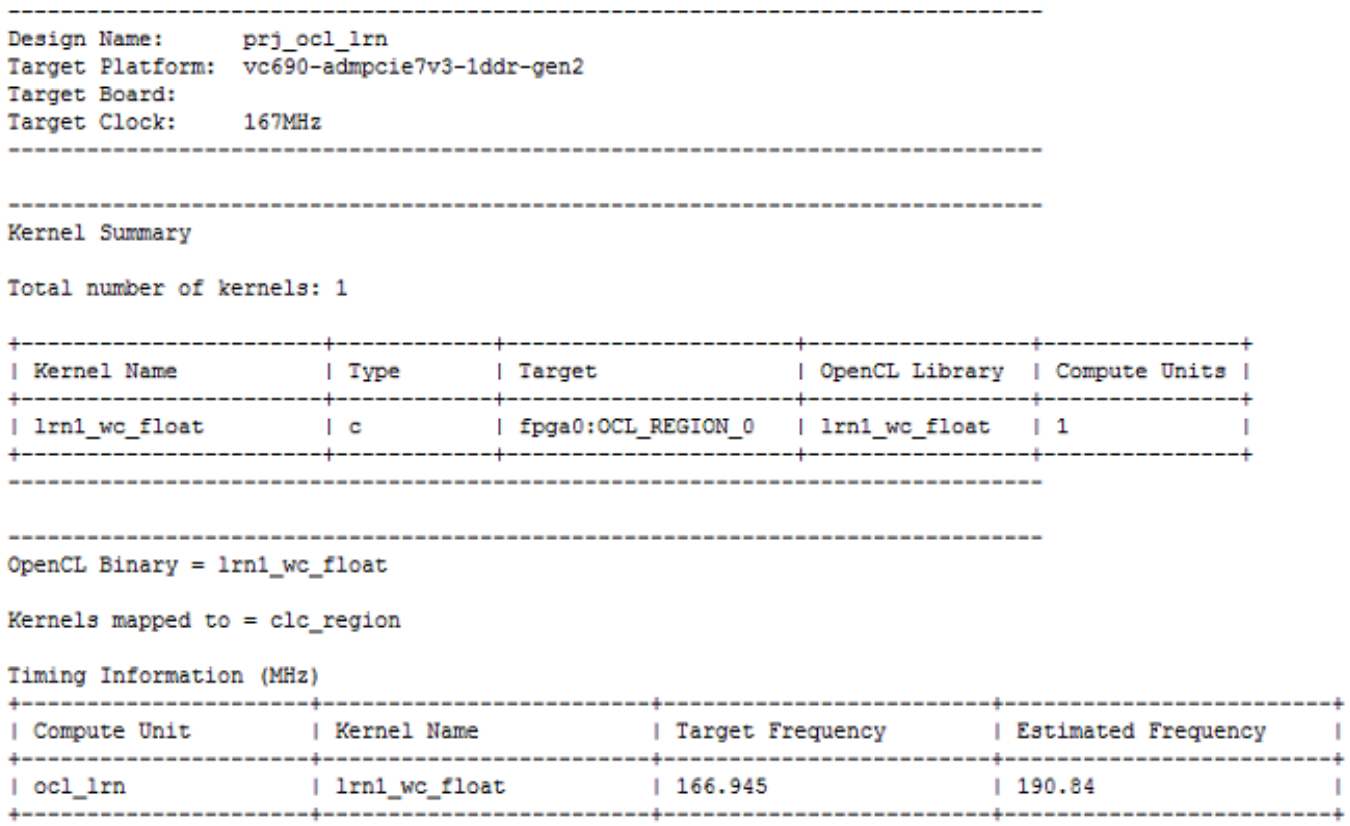
```
--------------------------------------------------------------------------------
Design Name:      prj_ocl_lrn
Target Platform:  vc690-admpcie7v3-1ddr-gen2
Target Board:
Target Clock:     167MHz
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Kernel Summary

Total number of kernels: 1

+-----------------------+------------+----------------------+-----------------+---------------+
| Kernel Name           | Type       | Target               | OpenCL Library  | Compute Units |
+-----------------------+------------+----------------------+-----------------+---------------+
| lrn1_ac_float         | c          | fpga0:OCL_REGION_0   | lrn1_ac_float   | 1             |
+-----------------------+------------+----------------------+-----------------+---------------+
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
OpenCL Binary = lrn1_ac_float

Kernels mapped to = clc_region

Timing Information (MHz)
+----------------------+-------------------------+-------------------------+-------------------------+
| Compute Unit         | Kernel Name             | Target Frequency        | Estimated Frequency     |
+----------------------+-------------------------+-------------------------+-------------------------+
| ocl_lrn              | lrn1_ac_float           | 166.945                 | 190.84                  |
+----------------------+-------------------------+-------------------------+-------------------------+
```

LRN Latency and Area information (AC)



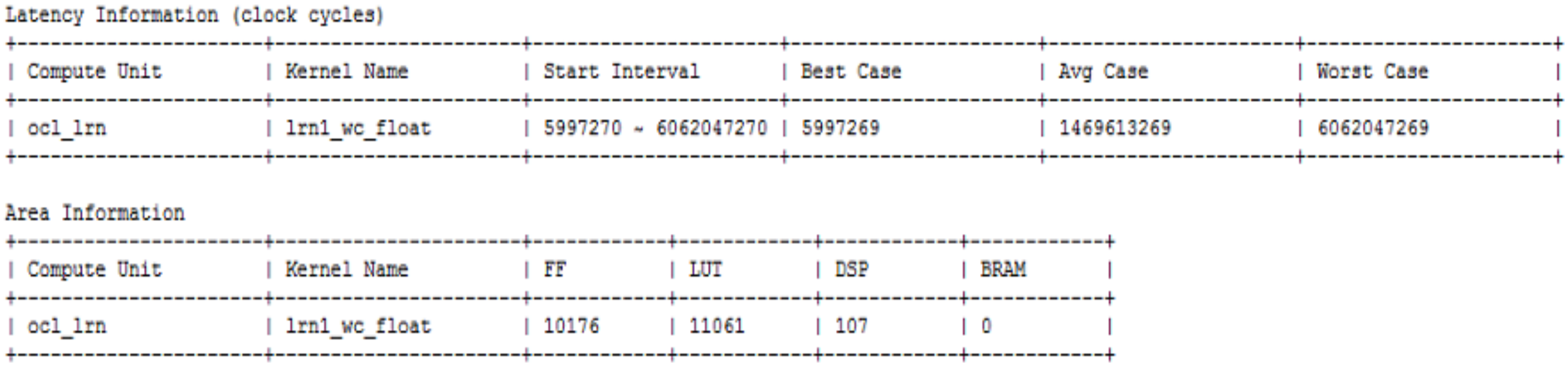
```
Latency Information (clock cycles)
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| Compute Unit        | Kernel Name         | Start Interval      | Best Case           | Avg Case            | Worst Case          |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| ocl_lrn             | lrn1_ac_float       | 6352502 ~ 9982502   | 6352501             | 9075001             | 9982501             |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+

Area Information
+---------------------+---------------------+-----------+-----------+-----------+-----------+
| Compute Unit        | Kernel Name         | FF        | LUT       | DSP       | BRAM      |
+---------------------+---------------------+-----------+-----------+-----------+-----------+
| ocl_lrn             | lrn1_ac_float       | 10244     | 10995     | 109       | 0         |
+---------------------+---------------------+-----------+-----------+-----------+-----------+
```

LRN Resource utilization and Timing information (WC)



```
------------------------------------------------------------------------------
Design Name:      prj_ocl_lrn
Target Platform:  vc690-admpcie7v3-1ddr-gen2
Target Board:
Target Clock:     167MHz
------------------------------------------------------------------------------

------------------------------------------------------------------------------
Kernel Summary

Total number of kernels: 1

+-----------------------+------------+----------------------+-----------------+---------------+
| Kernel Name           | Type       | Target               | OpenCL Library  | Compute Units |
+-----------------------+------------+----------------------+-----------------+---------------+
| lrn1_wc_float         | c          | fpga0:OCL_REGION_0   | lrn1_wc_float   | 1             |
+-----------------------+------------+----------------------+-----------------+---------------+
------------------------------------------------------------------------------

------------------------------------------------------------------------------
OpenCL Binary = lrn1_wc_float

Kernels mapped to = clc_region

Timing Information (MHz)
+----------------------+------------------------+------------------------+------------------------+
| Compute Unit         | Kernel Name            | Target Frequency       | Estimated Frequency    |
+----------------------+------------------------+------------------------+------------------------+
| ocl_lrn              | lrn1_wc_float          | 166.945                | 190.84                 |
+----------------------+------------------------+------------------------+------------------------+
```

LRN Latency and Area information (WC)



```
Latency Information (clock cycles)
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
| Compute Unit       | Kernel Name        | Start Interval     | Best Case          | Avg Case           | Worst Case         |
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
| ocl_lrn            | lrn1_wc_float      | 5997270 ~ 6062047270 | 5997269          | 1469613269         | 6062047269         |
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+

Area Information
+--------------------+--------------------+-----------+-----------+-----------+-----------+
| Compute Unit       | Kernel Name        | FF        | LUT       | DSP       | BRAM      |
+--------------------+--------------------+-----------+-----------+-----------+-----------+
| ocl_lrn            | lrn1_wc_float      | 10176     | 11061     | 107       | 0         |
+--------------------+--------------------+-----------+-----------+-----------+-----------+
```

Relu Layer

- Given an input value x, the ReLU layer computes the output as x if x > 0 and negative_slope * x if x <= 0.
- When the negative slope parameter is not set, it is equivalent to the standard ReLU function of taking max (x, 0).
- It also supports in-place computation, meaning that the bottom and the top blob could be the same to preserve memory consumption. i.e. (OWIDTH = IWIDTH and OHEIGHT = IHEIGHT)

Relu layer (contd ...)

- As per Alex net model, a Relu with max size of 50 x 50 is required.
- For Alex net’s requirement, the Relu function is standard max (x,0) function.
- 7 instances used in Alex net implementation.
- Pipelining is used for increasing throughput/ reducing overall latency.

Relu Resource utilization and Timing information

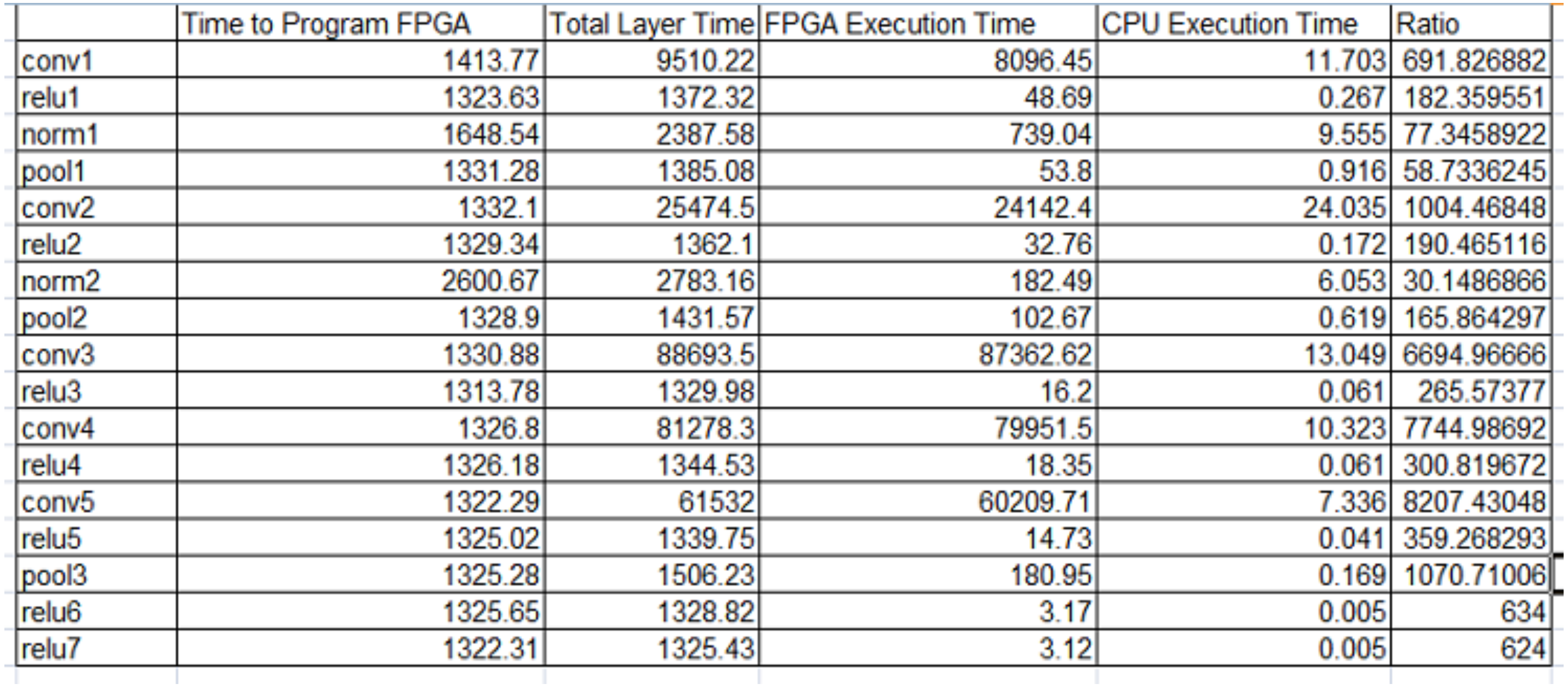
```
Kernel Summary:
Total number of kernels: 1
+----------------------+-----------+-------------------+----------------+--------------+
| Kernel Name          | Type      | Target            | OpenCL Library | Compute Units |
+----------------------+-----------+-------------------+----------------+--------------+
| relu_10              | c         | fpga0:OCL_REGION_0 | bin_relu_10    | 1            |
+----------------------+-----------+-------------------+----------------+--------------+
OpenCL Binary = bin_relu_10
Kernels mapped to = clc_region
Timing Information (MHz)
+---------------------+-----------------------+-----------------------+-----------------------+
| Compute Unit        | Kernel Name           | Target Frequency      | Estimated Frequency   |
+---------------------+-----------------------+-----------------------+-----------------------+
| relu_10             | relu_10               | 166.945               | 190.84                |
+---------------------+-----------------------+-----------------------+-----------------------+
```

Relu Latency information and Area utilization

```
Latency Information (clock cycles)
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| Compute Unit        | Kernel Name         | Start Interval      | Best Case           | Avg Case            | Worst Case          |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+
| relu_10             | relu_10             | 3038                | 3037                | 3037                | 3037                |
+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+

Area Information
+---------------------+---------------------+-----------+-----------+-----------+-----------+
| Compute Unit        | Kernel Name         | FF        | LUT       | DSP       | BRAM      |
+---------------------+---------------------+-----------+-----------+-----------+-----------+
| relu_10             | relu_10             | 881       | 1062      | 0         | 0         |
+---------------------+---------------------+-----------+-----------+-----------+-----------+
```

Roberto’s Demo

Performance comparison with CPU



	Time to Program FPGA	Total Layer Time	FPGA Execution Time	CPU Execution Time	Ratio
conv1	1413.77	9510.22	8096.45	11.703	691.826882
relu1	1323.63	1372.32	48.69	0.267	182.359551
norm1	1648.54	2387.58	739.04	9.555	77.3458922
pool1	1331.28	1385.08	53.8	0.916	58.7336245
conv2	1332.1	25474.5	24142.4	24.035	1004.46848
relu2	1329.34	1362.1	32.76	0.172	190.465116
norm2	2600.67	2783.16	182.49	6.053	30.1486866
pool2	1328.9	1431.57	102.67	0.619	165.864297
conv3	1330.88	88693.5	87362.62	13.049	6694.96666
relu3	1313.78	1329.98	16.2	0.061	265.57377
conv4	1326.8	81278.3	79951.5	10.323	7744.98692
relu4	1326.18	1344.53	18.35	0.061	300.819672
conv5	1322.29	61532	60209.71	7.336	8207.43048
relu5	1325.02	1339.75	14.73	0.041	359.268293
pool3	1325.28	1506.23	180.95	0.169	1070.71006
relu6	1325.65	1328.82	3.17	0.005	634
relu7	1322.31	1325.43	3.12	0.005	624

Testing of the layers

- Individually with raw data
- By Roberto using Caffe test vectors

Future Work

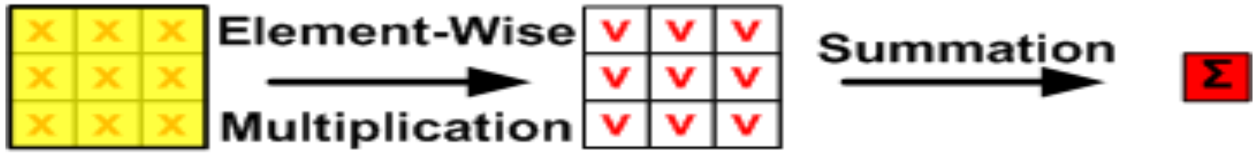
- 70 % percent Alex net is in hardware
- Target to convert it to 100 % hardware implementation.
- Fine tuning of individual layers to reduce latency

Future Work (Contd…)

- Placing multiple binary streams simultaneously on FPGA to reduce the fixed overhead.
- Measure performance over large number of images to reduce the effect of the fixed overhead

Future Work (Contd…)

- In Convolution, separate between element-wise multiplication step and results summation.
- This Enables the use of DataFlow directive.
- It requires more memory to store intermediate results.

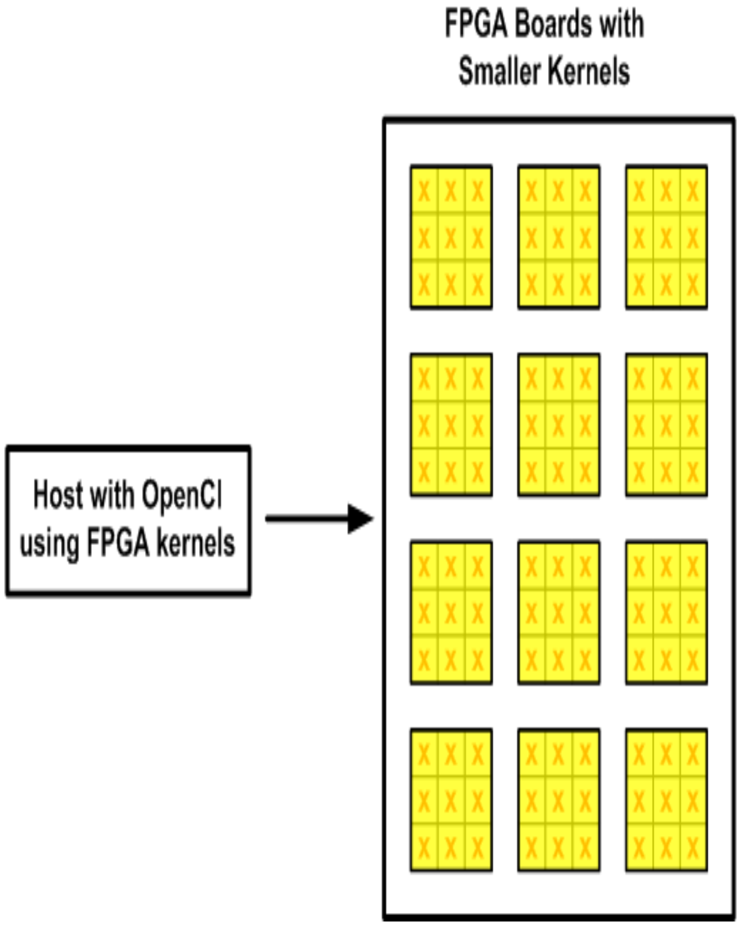


Future Work (Cont.)

	Input size	Filter size	Number of input features	Number of output features	Memory required for Synapses for one operation	Number of computations for one input set
Multi-Object recognition in natural images (DNN), winner 2012 ImageNet competition [2]	224	11	1	96	22.69 MB	Mult: ≈ 1.581 Billion Add: ≈ 1.567 Billion Division: ≈ 13.066 Mil Transfer: ≈ 13.066 Mil
Street scene parsing (CNN) (e.g., identifying building, vehicle, etc) [3]	500*375	9	32	48	0.24 MB	Mult: ≈ 67.39 Billion Add: ≈ 67.36 Billion Division: ≈ 26 Million Transfer: ≈ 26 Million
Face Detection in YouTube videos (DNN), (Google) [4]	200	18	8	8	1.29GB	N/A

Future Work (Contd…)

- In Convolution and Pooling, make hardware computation for only one window.
  - Reduce hardware resources needed
  - Increase parallelism


FPGA Boards with Smaller Kernels
Host with OpenCl using FPGA kernels