Statement of Work

Team: Neural Network – Theta
Members: Thibaut Lanois, Wen Wen, Jieming Pan

Feb 6, 2017

Scope of Work

The scope of work for Team Theta CNN on FPGA Project includes all proposing, planning, and implementation of a convolutional neural network on FPGA. We will be responsible for the design of a typical CNN on FPGA. We aim to implement LeNet-5-like CNN with two convolution, two pooling and multiple fully connected layers using pre-trained weights of the MNIST datasets. Specific deliverables and milestones will be listed in the Schedules and Milestones sections of this statement of work.

Work Requirements

As part of CNN on FPGA Project, Team Theta is responsible for implementing LeNet-5 on FPGA with help of the CPU. The following is a list of these tasks:

Kickoff:

- Presentation of project idea
- Scheduling and Statement of work
- Research on CNN and possible FPGA implementation

Software Design Phase:

- Implementing Convolution layer functions in C
- Implementing Pooling layer functions in C
- Implementing Fully connected layer functions in C
- Sigmoid function + Gaussian function
- Obtain weights from full LeNet-5
- Weight extraction
- Division into batch
- Finite state machine
- Image input to matrix
- Output display
- Testing

Hardware Design Phase:

- Implementing Convolution layer functions
- Implementing Pooling layer functions
- Implementing Fully connected layer functions
- Division into batch
- Finite state machine
- Sigmoid function in LUT
- Axi bus
- Router DSP register
- Router buffer - Block memory
- Integration

- Route placement
- Synthesis
- Testing

Schedule

<table>
<tr><th></th><th>30-Jan</th><th>6-Feb</th><th>13-Feb</th><th>20-Feb</th><th>27-Feb</th><th>6-Mar</th><th>13-Mar</th><th>20-Mar</th></tr>
<tr><td rowspan="6">Software Development</td><td colspan="2">Convolution layer in C</td><td colspan="3" rowspan="3">Testing</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Pooling layer in C</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Fully connecting layer in C</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Training in Matlab</td><td colspan="2">Weight extraction</td><td>Testing</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Divide into batch</td><td colspan="2">Controller</td><td>Testing</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="3">Image to matrix</td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="4">Hardware development</td><td></td><td colspan="3">Convolution layer and Pooling layer on DSP Slice</td><td colspan="4">Test bench</td></tr>
<tr><td></td><td colspan="3">Sigmoid function in LUT & Fully connected</td><td colspan="2">Router DSP register</td><td>Integration</td><td>Synthese + Place routes</td></tr>
<tr><td colspan="2">Research Router & Controller</td><td colspan="2">Axi Bus</td><td colspan="2">Router buffer – Block memory</td><td colspan="2"></td></tr>
<tr><td></td><td></td><td></td><td colspan="4">FSM ++</td><td></td></tr>
</table>

Milestones

The milestone dates are based on project presentations and phase-wise deliverables. To be specific:

- Feb 27: Software development is completed. The functional units designs of convolutional layers, pooling layers, sigmoid functions are completed.
- Mar 20: Integration of hardware design is completed.
- April 3: Testing for software and hardware designs are completed, and ready for baseline demo.
- April 17: Advanced features are completed.

Acceptance Criteria

This section of the Statement of Work defines how the customer will accept the deliverables resulting from this SOW. The acceptance of deliverables must be clearly defined and understood by all parties. This section should include a description of how both parties will know when work is acceptable, how it will be accepted, and who is authorized to accept the work.

Accuracy:

- Minimum: To prove that each layer works properly, we will compare the result of our C output with VHDL output using the same input and weight. If the difference in the result is less than ε, it is deemed as acceptable. $\varepsilon <$ C output - VHDL output, $\varepsilon = 0.0625$ (for 8 bit floating point)
- Optimal (advance features) : 85% accuracy

Throughput:

- Minimum: 500ms @ 100Mhz
- Optimal: 50ms @ 100Mhz