Project Plan report format – Your first page MUST match this format /10

Name: Nirmal Kumar Sedhumadhavan
Unityid: nsedhum@ncsu.edu
StudentID: 200483323

Summary Risk Plan:
1. Meeting all the timing constraints throughout the whole logic design.
2. To store all obtained intermediate results in synchronized manner into the register and the scratch SRAM.
3. Writing the output into SRAM in sequential addresses correspondingly.
4. Down Counter Implementation for size counts promptly throughout the logic design.
5. Achieving desired output after each stage of the neural network.
6. Synthesizable design for all combinational logic.
7. Achieving minimal slack time and minimal area during synopsys.

Schedule:
1. 10/29/2022 – High level logic design and plan.
2. 11/02/2022 – Detailed logic design with appropriate data and control path.
3. 11/08/2022 – Pseudocode, FSD &Timing Diagram
4. 11/14/2022 – Verilog code & synthesis script.

Brief Description of Mode of operation, including selected algorithms

1. For address reading using 8*3 Mux with one line as a counter to increment the address location and with one line for using the same previous value, and other lines with combinational logic for iterating through the matrix to access 3*3 matrix for convolution.
2. Same way accessing address from weight matrix using a mux to iterate sequentially (column wise) in the given 3*3 kernel matrix.
3. Use of Mux and a Flip-Flop register to separate 2 inputs located in same address and storing in register to use for convolution with weight matrix.
4. Implementation of Size counter to down count the number input (9 elements) and a convolution counter to down count the total number of convolutions count in the I/P matrix ((N-2) * (N-2)). All counters are implemented using mux with a decrement subtractor.
5. Multiplication is done using 8*8 bit signed multiplier 9 times every cycle as there are 9 elements to be multiplied in 3*3 matrix and accumulated using accumulator.
6. Use of Relu Activation Function to saturate value above 127 using 8-bit signed comparator and storing the convolution matrix in scratch SRAM.
7. Max pooling is done with a mux to iterate through convolution matrix as a 2*2 matrix and then storing maximum of the 2*2 matrix in a Flip-Flop and latching it again in scratch SRAM. Generated max pooling matrix size is (N-2)/2 * (N-2)/2.
8. For address reading from max pooling matrix stored in scratch SRAM, use of 8*3 Mux with one line as a counter to increment the address location and with one line for using the same previous value, and other lines with combinational logic for iterating through the matrix to access 3*3 matrix for fully connected layer.
9. Use of Mux and a Flip-Flop register to separate 2 inputs located in same address and storing in register.
10. Same way accessing address from weight matrix using a mux to iterate sequentially, but this time not column wise rather row wise in the given 3*3 kernel matrix.
11. Implementation of Size counter to down count the number of fully connected count in the max pooling matrix ((N-2)/2 * (N-2)/2).
12. Multiplication is done using 8*8 bit signed multiplier 9 times every cycle as there are 9 elements to be multiplied in 3*3 matrix and accumulated using accumulator.
13. Use of Relu Activation Function to saturate value above 127 using 8-bit signed comparator and storing the final fully connected matrix in output SRAM.

High level sketch. Add details on the following pages if necessary