EE577B
Lab4
Report

Huayu Fu
4745159848

1. Design idea:
   The design supports **configure programmable DNN**, by changing the parameters in the tb file. The default is 4-3-5, dnn.
   The input of DNN and the output of each neuron are connected to a global register core_dataout.
   There is a state machine inside the mcp(the whole dnn machine) module. it will start the calculation layer by layer.
   The start signal will start the computation.
   The w value is a data bus connected to the test bench.
   First it will connect the first output from the layer 1 neuron 1 to all layer 2 module, and assign the w value corresponding to the layer 1 neuron 1 to layer 2 neuron 1-n. then have the mux select the output from the neuron 2. And so on.
   After the first layer finished, the second layer computation started.
   Until the last layer. then there will be a ready signal sends out. And it will send all calculation result from the last layer to output, clock by clock, each clock will out put one data from last layer neuron. 1-n
2. Explanation of each module:
   Mcp is the whole dnn network. And the size is configurable by the parameter.
   Neuron is the individual neuron. It have input of data and w and the output ports.
   It have one register called value.
   Value is initially to be 0. And when mode==1(inner product mode) value = data*w+value
   When mode==2(inner sigmoid activation) using the LUT.
   When mode==0(NOOP)
3. Look up table:

```
case(mode)
  1: value<=value+{in,{(16-width){1'b0}}}*{w,{(16-width){1'b0}}};
  2: begin
    if(value>500000) value <= 65535;
    else if(value>250000) value <= 64869;
    else if(value>150000) value <= 62577;
    else if(value>97500) value <= 53831;
    else if(value>92500) value <= 53079;
    else if(value>87500) value <= 52292;
    else if(value>82500) value <= 51467;
    else if(value>77500) value <= 50606;
    else if(value>72500) value <= 49708;
    else if(value>67500) value <= 48774;
    else if(value>62500) value <= 47805;
    else if(value>57500) value <= 46801;
    else if(value>52500) value <= 45764;
    else if(value>47500) value <= 44695;
    else if(value>42500) value <= 43595;
    else if(value>37500) value <= 42468;
    else if(value>32500) value <= 41316;
    else if(value>27500) value <= 40139;
    else if(value>22500) value <= 38943;
    else if(value>17500) value <= 37729;
    else if(value>12500) value <= 36501;
    else if(value>7500) value <= 35263;
    else if(value>2500) value <= 34017;
    else value <= 32768;
  end
  default: value <= value;
dcase
```

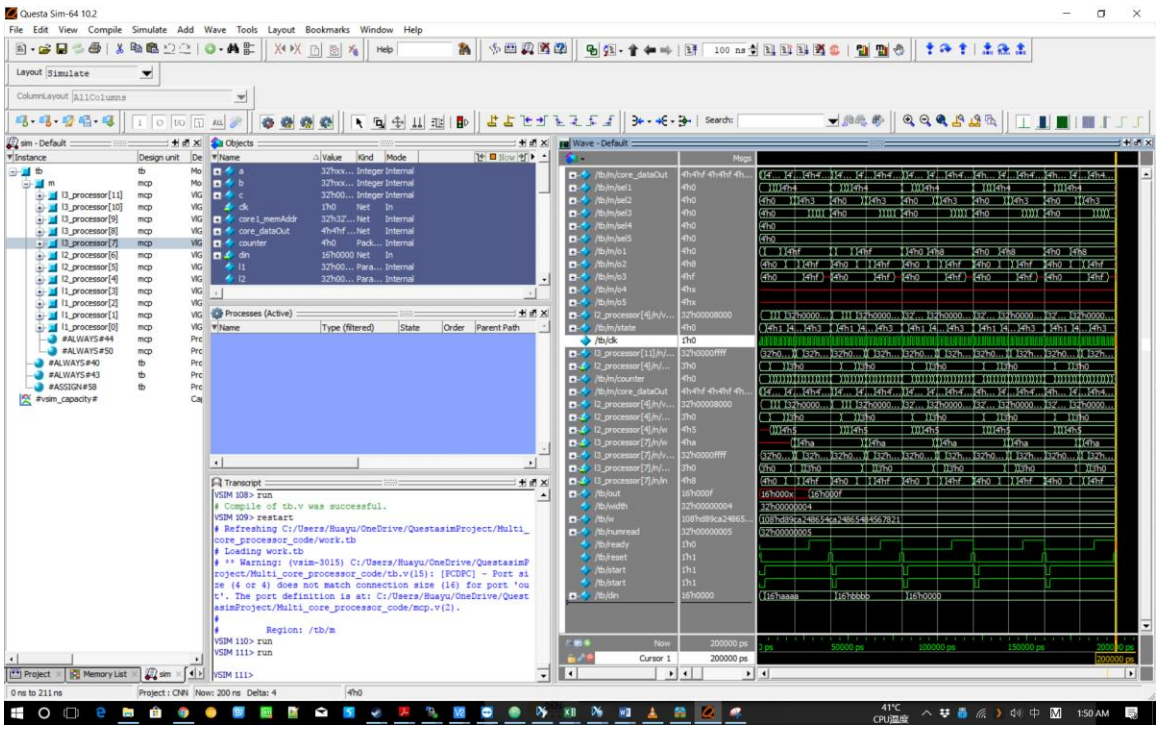
4. Size configurable dnn code

```
genvar i;
generate
  for (i=0; i<l1; i=i+1) begin :l1_processor
    assign core_dataOut[i]=din[(i+1)*width-1-:width];
  end
  for (i=l1; i<l1+l2; i=i+1) begin :l2_processor
    neuron #(width) n(clk,reset,mode[0],o1,w[i],core_dataOut[i]);
  end
  for (i=l1+l2; i<l1+l2+l3; i=i+1) begin :l3_processor
    neuron #(width) n(clk,reset,mode[1],o2,w[i],core_dataOut[i]);
  end
  for (i=l1+l2+l3; i<l1+l2+l3+l4; i=i+1) begin :l4_processor
    neuron #(width) n(clk,reset,mode[2],o3,w[i],core_dataOut[i]);
  end
  for (i=l1+l2+l3+l4; i<total; i=i+1) begin :l5_processor
    neuron #(width) n(clk,reset,mode[3],o4,w[i],core_dataOut[i]);
  end
endgenerate
```

5. Select mux for feeding data to neuron

```
always@(core_dataOut,sel1,sel2,sel3,sel4,sel5) begin
  if(l1>0) o1<=core_dataOut[sel1];
  if(l2>0) o2<=core_dataOut[l1+sel2];
  if(l3>0) o3<=core_dataOut[l1+l2+sel3];
  if(l4>0) o4<=core_dataOut[l1+l2+l3+sel4];
end
```

6. Simulation



7. Demonstrate for configurable size