module softmax(output [19:0]out1,[19:0]out2,[19:0]out3,[19:0]out4,input [19:0]In1,input [19:0]In2,input [19:0]In3,input [19:0]In4);

wire [19:0] Ex1,Ex2,Ex3,Ex4;

wire [19:0] Est3,Xo,Acc,Est1,Est2;

wire [21:0]T2;

wire[21:0] Acc\_temp;

wire [40:0]T1,o1,o4,out\_1,out\_2,out\_3,out\_4,k1,k2,o11,o12,o41,o42;

wire [1:0]i;//shift

//calculated exponential

pex p1(Ex1,In1);

pex p2(Ex2,In2);

pex p3(Ex3,In3);

pex p4(Ex4,In4);

//denominator

assign Acc\_temp=Ex2+Ex3+Ex4+Ex1;

assign Acc=Acc\_temp;

//normalization

shift s1(i,Acc);

//initial value

assign T1=(20'd124957)\*(Acc>>i);

assign T2=T1[35:16];

assign Xo=20'd184569-T2;

//first iteration

NR N1(Xo,i,Acc,Est1);

NR N2(Est1,i,Acc,Est2);

NR N3(Est2,i,Acc,Est3);

//multiply with numerator

assign out\_1=(Ex1>>i)\*Est3;

assign out1=out\_1[35:16];

assign out\_2=(Ex2>>i)\*Est3;

assign out2=out\_2[35:16];

assign out\_3=(Ex3>>i)\*Est3;

assign out3=out\_3[35:16];

assign out\_4=(Ex4>>i)\*Est3;

assign out4=out\_4[35:16];

endmodule

module pex(output [27:0]Exp,input [19:0]x);

wire [43:0]squarex;

wire[45:0]cubex;

wire [46:0]quadx;

wire [46:0] temp3,temp2,temp1;

wire [27:0]square,out1;

wire [29:0]cube,out2;

wire [30:0]quad,out3;

wire [43:0]out;

assign squarex=x\*x;

assign square=squarex[35:16];

assign cubex=square\*x;

assign cube=cubex[35:16];

assign quadx=cube\*x;

assign quad=quadx[46:16];

assign temp3=quad\*(20'd2730);

assign out3=temp3[46:16];

assign temp2=cube\*(20'd10922);

assign out2=temp2[45:16];

assign temp1=square\*(20'd32768);

assign out1=temp1[42:16];

assign out=20'd65536+x+out1+out2+out3;

assign Exp=out[27:0];

endmodule

module shift(output reg [0:1] shift,input [19:0]Den);

wire [19:0]temp=Den>>1;

wire [19:0]temp1=Den>>2;

wire [19:0]temp2=Den>>3;

wire [19:0]temp3=Den>>4;

//wire [19:0]r4=Den>>5;

//wire [19:0]r5=Den>>6;

//wire [19:0]r6=Den>>7;

always@(\*)

begin

if(20'd32768<temp && temp<20'd65536)

shift=1;

else if(20'd32768<temp1 && temp1<20'd65536)

shift=2;

else if(20'd32768<temp2 &&temp2<20'd65536)

shift=3;

else if(20'd32768<temp3 && temp3<20'd65536)

shift=4;

//else if(20'd32768<r4 && temp3<20'd65536)

//shift=5;

//else if(20'd32768<r5 && temp3<20'd65536)

//shift=6;

//else if(20'd32768<r6 && temp3<20'd65536)

//shift=7;

else

shift=0;

end

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 03/26/2024 09:02:59 PM

// Design Name:

// Module Name: NR

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module NR(input [19:0]X,input [1:0]shift, input [19:0]D, output [19:0]Xi);

wire [19:0]o21,o31;

wire [40:0]o11,o41,o51;

assign o11=(D>>shift)\*X;

assign o21=o11[35:16];

assign o31=20'd131072-o21;

assign o41=X\*o31;

assign Xi=o41[35:16];

endmodule

module Soft\_tb();

reg [19:0]x1,x2,x3,x4;

wire [19:0]ya,yb,yc,yd;

softmax e1(ya,yb,yc,yd,x1,x2,x3,x4);

initial begin

x1=20'd65536;

x2=20'd32768;

x3=20'd16384;

x4=20'd8192;

end

endmodule