何昊天10月、11月工作汇报

论文题目：求解二次规划问题的基于LVI的原-对偶神经网络FPGA设计和实现

论文作者：袁银娟

论文链接：http://www.wanfangdata.com.cn/details/detail.do?_type=degree&id=Y1085686

基于LVI（Linear Variational Inequalities）的原-对偶神经网络（Primal-Dual Neural Network，PDNN）可以用来求解线性规划和同时含有等式约束、不等式约束和界限约束（激活函数fuction分段线性）的二次规划问题。PDNN实质上是一类RNN（Recurrent Neural Network），并对PDNN网络在纯FPGA上的实现做出贡献。

一、网络设计

二次规划问题的标准形式为：（W为半正定型）

$$\begin{array}{ll} minimize & x^T Wx/2 + q^T x; \\ subject\ to & Jx = d, Ax \le b, \varepsilon^- \le x \le \varepsilon^+ \end{array} \tag{1}$$

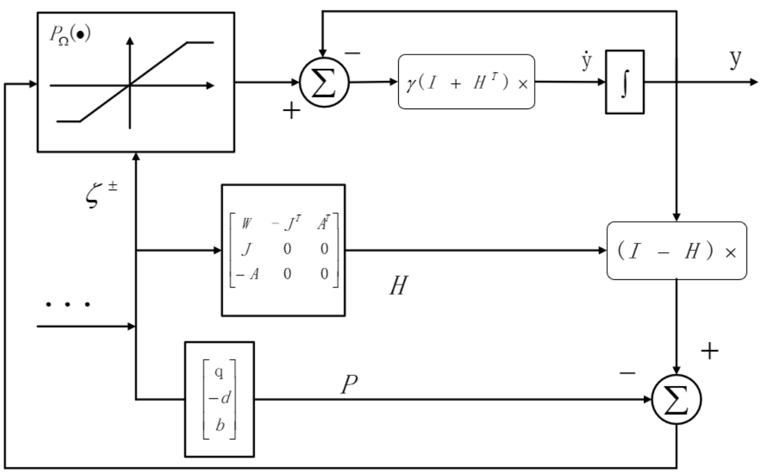
经原文推导，一般二次规划问题可以转化为基于LVI的 原-对偶神经网络动态方程：

$$\dot{y} = \gamma(1 + H^T)\{P_\Omega(y - (Hy + p)) - y\} \tag{2}$$

其中：

$$H = \begin{bmatrix} W & -J^T & A^T \\ J & 0 & 0 \\ -A & 0 & 0 \end{bmatrix}, \qquad p = \begin{bmatrix} q \\ -d \\ b \end{bmatrix} \tag{3}$$

故可设计PDNN框图：



维数越高，硬件实现越复杂，FPGA上存储资源十分有限。这也是神经网络在纯FPGA 上实现的难点与障碍。本文研究简单的二维情况，故做以下赋值定义：

$$W = \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix}, \qquad q = [q_1 \quad q_2], \qquad j = [j_1 \quad j_2], \qquad d = d, \tag{4}$$

$$a = [a_1 \quad a_2], \qquad b = b, \qquad \varepsilon^- = \begin{bmatrix} \varepsilon_1^- \\ \varepsilon_2^- \end{bmatrix}, \qquad \varepsilon^+ = \begin{bmatrix} \varepsilon_1^+ \\ \varepsilon_2^+ \end{bmatrix}$$

结合前面的式子可得：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{u} \\ \dot{v} \end{bmatrix} = \gamma \begin{bmatrix} 1+\omega_{11} & \omega_{21} & j_1 & -a_1 \\ \omega_{12} & 1+\omega_{22} & j_2 & -a_2 \\ -j_1 & -j_2 & 1 & 0 \\ a_1 & a_2 & 0 & 1 \end{bmatrix} \left\{ P_\Omega \left(\begin{bmatrix} 1-\omega_{11} & -\omega_{12} & j_1 & -a_1 \\ -\omega_{12} & 1-\omega_{22} & j_2 & -a_2 \\ -j_1 & -j_2 & 1 & 0 \\ a_1 & a_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \\ v \end{bmatrix} - \begin{bmatrix} q_1 \\ q_2 \\ -d \\ b \end{bmatrix} \right) - \begin{bmatrix} x_1 \\ x_2 \\ u \\ v \end{bmatrix} \right\} \tag{5}$$

化为代数式为：

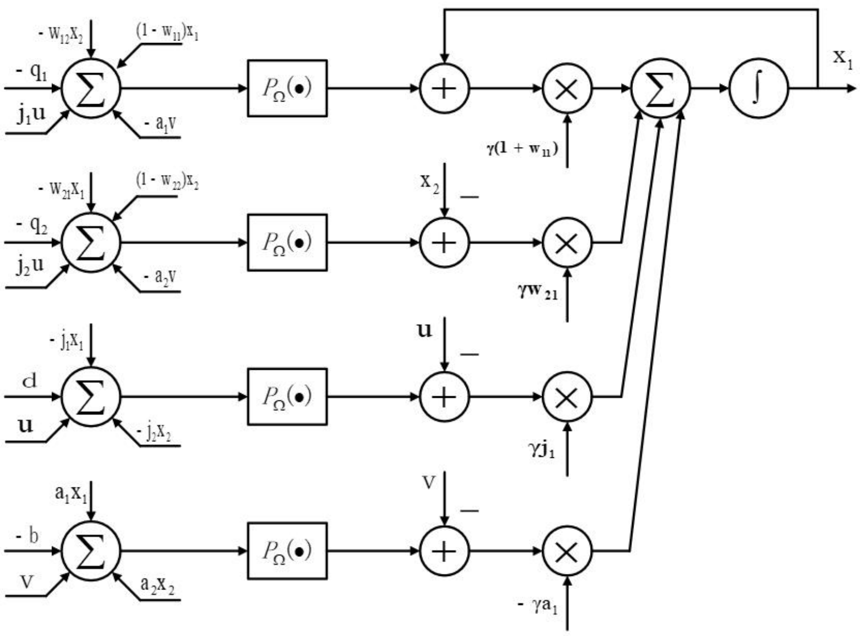
$$
\begin{aligned}
\dot{x_1} =&\gamma(w_{11}+1)\left\{P_\Omega[(1-w_{11})x_1+(-w_{12})x_2+j_1u+(-a_1)v-q_1]-x_1\right\}\\
&+\gamma w_{21}\left\{P_\Omega[(-w_{21})x_1+(1-w_{22})x_2+j_2u+(-a_2)v-q_2]-x_2\right\}\\
&+\gamma j_1\left\{P_\Omega[(-j_1)x_1+(-j_2)x_2+u+d]-u\right\}\\
&+(-\gamma a_1)\left\{P_\Omega[a_1x_1+a_2x_2+v-b]-v\right\}
\end{aligned}
\tag{6-1}
$$

$$
\begin{aligned}
\dot{x_2} =&\gamma w_{12}\left\{(P_\Omega[(1-w_{11})x_1+(-w_{012})x_2+j_1u+(-a_1)v-q_1]-x_1)\right\}\\
&+\gamma(1+w_{022})\left\{(P_\Omega[(-w_{021})x_1+(1-w_{022})x_2+j_2u+(-a_2)v-q_2]-x_2)\right\}\\
&+\gamma j_2\left\{(P_\Omega[(-j_1)x_1+(-j_2)x_2+u+d]-u)\right\}\\
&+(-\gamma a_2)\left\{(P_\Omega[a_1x_1+a_2x_2+v-b]-v)\right\}
\end{aligned}
\tag{6-2}
$$

$$
\begin{aligned}
\dot{u} =&\gamma(-j_1)\left\{(P_\Omega[\{(1-w_{11})x_1+(-w_{012})x_2+j_1u+(-a_1)v-q_1\}]-x_1)\right\}\\
&+\gamma(-j_2)\left\{(P_\Omega[\{(-w_{21})x_1+(1-w_{22})x_2+j_2u+(-a_2)v-q_2\}]-x_2)\right\}\\
&+\gamma\left\{(P_\Omega[(-j_1)x_1+(-j_2)x_2+u+d]-u)\right\}
\end{aligned}
\tag{6-3}
$$

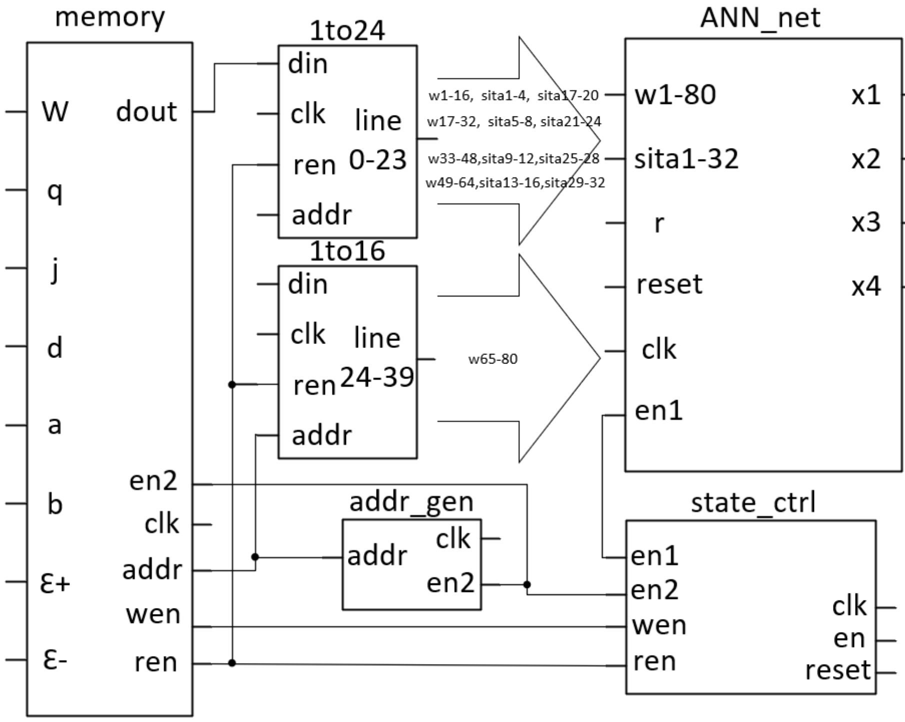
$$
\begin{aligned}
\dot{v} =&\gamma a_1\left\{P_\Omega[\{(1-w_{11})x_1+(-w_{12})x_2+j_1u+(-a_1)v-q_1\}]-x_1\right\}\\
&+\gamma a_2\left\{P_\Omega[\{(-w_{21})x_1+(1-w_{22})x_2+j_2u+(-a_2)v-q_2\}]-x_2\right\}\\
&+\gamma\left\{P_\Omega[a_1x_1+a_2x_2+v-b]-v\right\}
\end{aligned}
\tag{6-4}
$$

根据代数式（6-1）可设计x_1的模块框图：



同理由代数式（6-2~4）可设计x_2、u、v等模块框图。

针对上述研讨，设计如下PDNN的top模块：



参数由存储模块（memory）左侧的输入端口输入，网络模块（ANN_net）的 γ 口输入。

状态控制模块（state_ctrl）控制整个网络的时序逻辑，其clk为公共的，en仅控制该模块的使能情况，其输出en1为网络模块（ANN_net）使能输入，en2为地址模块（addr_gen）以及memory模块的使能输入。

下面叙述整个网络的工作流程：【state_ctrl=（en1,en2,wen,ren）】

a.初始状态：（en1,en2,wen,ren）=（1,1,1,1）

state_ctrl模块停止工作，网络停止工作。

b.写状态：（en1,en2,wen,ren）=（1,0,0,1）

memory模块写操作使能，计算40个基础参数。

c.读状态：（en1,en2,wen,ren）=（1,0,1,0）

memory模块读操作使能，并按addr_gen模块生成的地址码依次写入分配模块（1to24、1to16）使能，将40个基础参数写入ANN_net中。

d.计算状态：（en1,en2,wen,ren）=（0,1,1,1）

ANN计算并输出x_1、x_2、x_3、x_4即为x_1、x_2、u、v。

二、参数计算并验证

输入参数后的ANN_net是网络的计算部分，其内部多次反馈x_{1-4}的输出，并经过一系列调用加减乘以及激活函数等模块（note_adder、note_acc、note_sub、mult_32x32、fuction1、fuction2、fuction3），并输入到积分模块（Integrator）进而得到网络计算的最终结果，下面是基于ANN_net内部细节的计算表达式：（其中h=γ，为设计参数）

$$\begin{gathered}\{(f_1[x_1w_1+x_2w_2+uw_3+vw_4-sita1,sita17,sita18]-x_1)*w_{65}\\+(f_1[x_1w_5+x_2w_6+uw_7+vw_8-sita2,sita19,sita20]-x_2)*w_{66}\\+(f_2[x_1w_9+x_2w_{10}+uw_{11}+vw_{12}-sita3]-u)*w_{67}\\+(f_3[x_1w_{13}+x_2w_{14}+uw_{15}+vw_{16}-sita4,]-v)*w_{68})\}\\\cdot h\overset{\int}{=}x_1\end{gathered}\tag{7-1}$$

$$\begin{gathered}\{(f_1[x_1w_{17}+x_2w_{18}+uw_{19}+vw_{20}-sita5,sita21,sita22]-x_1)*w_{69}\\+(f_1[x_1w_{21}+x_2w_{22}+uw_{23}+vw_{24}-sita6,sita23,sita24]-x_2)*w_{70}\\+(f_2[x_1w_{25}+x_2w_{26}+uw_{27}+vw_{28}-sita7]-u)*w_{71}\\+(f_3[x_1w_{29}+x_2w_{30}+uw_{31}+vw_{32}-sita8]-v)*w_{72}\}\\\cdot h\overset{\int}{=}x_2\end{gathered}\tag{7-2}$$

$$
\begin{aligned}
&\{(f_1[x_1w_{33}+x_2w_{34}+uw_{35}+vw_{36}-sita9,,sita25,sita26]-x_1)*w_{73}\\
&+(f_1[x_1w_{37}+x_2w_{38}+uw_{39}+vw_{40}-sita10,,sita27,sita28]-x_2)*w_{74}\\
&+(f_2[x_1w_{41}+x_2w_{42}+uw_{43}+vw_{44}-sita11]-u)*w_{75}\\
&+(f_3[x_1w_{45}+x_2w_{46}+uw_{47}+vw_{48}-sita12]-v)*w_{76}\}\\
&\cdot h \overset{\int}{=} u
\end{aligned}
\tag{7-3}
$$

$$
\begin{aligned}
&\{(f_1[x_1w_{49}+x_2w_{50}+uw_{51}+vw_{52}-sita13,sita29,sita30]-x_1)*w_{77}\\
&+(f_1[x_1w_{53}+x_2w_{54}+uw_{55}+vw_{56}-sita14,sita31,sita32]-x_2)*w_{78})\\
&+(f_2[x_1w_{57}+x_2w_{58}+uw_{59}+vw_{60}-sita15]-u)*w_{79})\\
&+(f_3[x_1w_{61}+x_2w_{62}+uw_{63}+vw_{64}-sita16]-v)*w_{80})\}\\
&\cdot h \overset{\int}{=} v
\end{aligned}
\tag{7-4}
$$

将memery计算后的40个参数分别输入后：

$$
\begin{aligned}
&\{(f_1[\{x_1\cdot(-|w_{011}-1|)+x_2\cdot(-w_{012})+u\cdot j_1+v\cdot(-a_1)-q_1\},e_1,e_2]-x_1)\cdot(|w_{011}+1|)\\
&+(f_1[\{x_1\cdot(-w_{021})+x_2\cdot(-|w_{022}-1|)+u\cdot j_2+v\cdot(-a_2)-q_2\},e_3,e_4]-x_2)\cdot w_{021}\\
&+(f_2[x_1\cdot(-j_1)+x_2\cdot(-j_2)+u\cdot 1+v\cdot 0-(-d)]-u)\cdot j_1\\
&+(f_3[x_1\cdot a_1+x_2\cdot a_2+u\cdot 0+v\cdot 1-b]-v)\cdot(-a_1)\}\\
&\cdot h \overset{\int}{=} x_1
\end{aligned}
\tag{8-1}
$$

$$
\begin{aligned}
&\{(f_1[\{x_1\cdot(-|w_{011}-1|)+x_2\cdot(-w_{012})+u\cdot j_1+v\cdot(-a_1)-q_1\},e_1,e_2]-x_1)\cdot w_{012}\\
&+(f_1[\{x_1\cdot(-w_{021})+x_2\cdot(-|w_{022}-1|)+u\cdot j_2+v\cdot(-a_2)-q_2\},e_3,e_4]-x_2)\cdot(|w_{022}+1|)\\
&+(f_2[x_1\cdot(-j_1)+x_2\cdot(-j_2)+u\cdot 1+v\cdot 0-(-d)]-u)\cdot j_2\\
&+(f_3[x_1\cdot a_1+x_2\cdot a_2+u\cdot 0+v\cdot 1-b]-v)\cdot(-a_2)\}\\
&\cdot h \overset{\int}{=} x_2
\end{aligned}
\tag{8-2}
$$

$$
\begin{aligned}
&\{(f_1[\{x_1\cdot(-|w_{011}-1|)+x_2\cdot(-w_{012})+u\cdot j_1+v\cdot(-a_1)-q_1\},e_1,e_2]-x_1)\cdot(-j_1)\\
&+(f_1[\{x_1\cdot(-w_{021})+x_2\cdot(-|w_{022}-1|)+u\cdot j_2+v\cdot(-a_2)-q_2\},e_3,e_4]-x_2)\cdot(-j_2)\\
&+(f_2[x_1\cdot(-j_1)+x_2\cdot(-j_2)+u\cdot 1+v\cdot 0-(-d)]-u)\cdot 1\\
&+(f_3[x_1\cdot a_1+x_2\cdot a_2+u\cdot 0+v\cdot 1-b]-v)\cdot 0\}\\
&\cdot h \overset{\int}{=} u
\end{aligned}
\tag{8-3}
$$

$$
\begin{aligned}
&\{(f_1[\{x_1\cdot(-|w_{011}-1|)+x_2\cdot(-w_{012})+u\cdot j_1+v\cdot(-a_1)-q_1\},e_1,e_2]-x_1)\cdot a_1\\
&+(f_1[\{x_1\cdot(-w_{021})+x_2\cdot(-|w_{022}-1|)+u\cdot j_2+v\cdot(-a_2)-q_2\},e_3,e_4]-x_2)\cdot a_2\\
&+(f_2[x_1\cdot(-j_1)+x_2\cdot(-j_2)+u\cdot 1+v\cdot 0-(-d)]-u)\cdot 0\\
&+(f_3[x_1\cdot a_1+x_2\cdot a_2+u\cdot 0+v\cdot 1-b]-v)\cdot 1\}\\
&\cdot h \overset{\int}{=} v
\end{aligned}
\tag{8-4}
$$

化简后：

$$
\begin{aligned}
&\{(P_\Omega[(1-w_{11})x_1+(-w_{12})x_2+j_1u+(-a_1)v-q_1]-x_1)\cdot(w_{11}+1)\\
&+(P_\Omega[(-w_{21})x_1+(1-w_{22})x_2+j_2u+(-a_2)v-q_2]-x_2)\cdot w_{21}\\
&+(P_\Omega[(-j_1)x_1+(-j_2)x_2+u+d]-u)\cdot j_1\\
&+(P_\Omega[a_1x_1+a_2x_2+v-b]-v)\cdot(-a_1)\}\\
&\cdot\gamma=\dot{x_1}\overset{\int}{=} x_1
\end{aligned}
\tag{9-1}
$$

$$
\begin{aligned}
&\{(P_\Omega[(1-w_{11})x_1+(-w_{012})x_2+j_1u+(-a_1)v-q_1]-x_1)\cdot w_{12}\\
&+(P_\Omega[(-w_{021})x_1+(1-w_{022})x_2+j_2u+(-a_2)v-q_2]-x_2)\cdot(w_{022}+1)\\
&+(P_\Omega[(-j_1)x_1+(-j_2)x_2+u+d]-u)\cdot j_2\\
&+(P_\Omega[a_1x_1+a_2x_2+v-b]-v)\cdot(-a_2)\}\\
&\cdot\gamma=\dot{x_2}\overset{\int}{=} x_2
\end{aligned}
\tag{9-2}
$$

$$
\begin{aligned}
&\{(P_\Omega[\{(1-w_{11})x_1+(-w_{012})x_2+j_1u+(-a_1)v-q_1\}]-x_1)\cdot(-j_1)\\
&+(P_\Omega[\{(-w_{21})x_1+(1-w_{22})x_2+j_2u+(-a_2)v-q_2\}]-x_2)\cdot(-j_2)\\
&+(P_\Omega[(-j_1)x_1+(-j_2)x_2+u+d]-u)\}\\
&\cdot\gamma=\dot{u}\overset{\int}{=}u
\end{aligned}
\tag{9-3}
$$

$$
\begin{aligned}
&\{(P_\Omega[\{(1-w_{11})x_1+(-w_{12})x_2+j_1u+(-a_1)v-q_1\}]-x_1)\cdot a_1\\
&+(P_\Omega[\{(-w_{21})x_1+(1-w_{22})x_2+j_2u+(-a_2)v-q_2\}]-x_2)\cdot a_2\\
&+(P_\Omega[a_1x_1+a_2x_2+v-b]-v)\}\\
&\cdot\gamma=\dot{v}\overset{\int}{=}v
\end{aligned}
\tag{9-4}
$$

经比较，（9）式即为（6）式，即本网络很好的实现了设计。

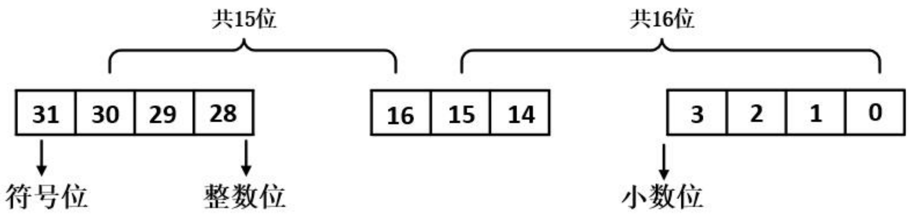
三、实例求解并验证

通过实例求解，将用上文实现的基于LVI的原-对偶神经网络进行验证，引入（1）式问题，并作如下设定：

$$
W=\begin{bmatrix}\omega_{11} & \omega_{12}\\ \omega_{21} & \omega_{22}\end{bmatrix}=\begin{bmatrix}2 & 2\\ 2 & 6\end{bmatrix},\quad q=[q_1\quad q_2]=[1\quad 1],\quad J=[j_1\quad j_2]=[1\quad 1],\quad d=d=1,\ (10)
$$

$$
a=[a_1\quad a_2]=[3\quad 4],\qquad b=b=5,\qquad \varepsilon^-=\begin{bmatrix}\varepsilon_1^-\\ \varepsilon_2^-\end{bmatrix}=\begin{bmatrix}-6\\ -6\end{bmatrix},\qquad \varepsilon^+=\begin{bmatrix}\varepsilon_1^+\\ \varepsilon_2^+\end{bmatrix}=\begin{bmatrix}6\\ 6\end{bmatrix}
$$

把各参数表示为32位定点数。最高位（第31位）为符号位，0表示正数，1表示负数；第30-16位（共15位）为整数位，第15-0（共16位）为小数位，如下图。并设定系统的定点数仿真步长$\gamma=1\times2^{-10}$



将（10）式输入到（1）式中，可得：

$$
\begin{aligned}
&minimize &&x_1^2+3x_2^2+2x_1x_2+x_1+x_2;\\
&subject\ to &&x_1+x_2=1,3x_1+x_2\le 5,-6\le x\le 6
\end{aligned}
\tag{11}
$$

将上式输入到matlab中尝试求解，代码如下：

```
>> clc
>> clear
>> W = [2,2;2,6];
>> q = [1;1];
>> J = [1,1];
>> d = 1;
>> A = [3,4];
>> b = 5;
>> lb = [-6;-6];
>> ub = [6;6];
>> [x,fval,exitflag,output,lambda]=quadprog(W,q,A,b,J,d,lb,ub)
```

结果如下：

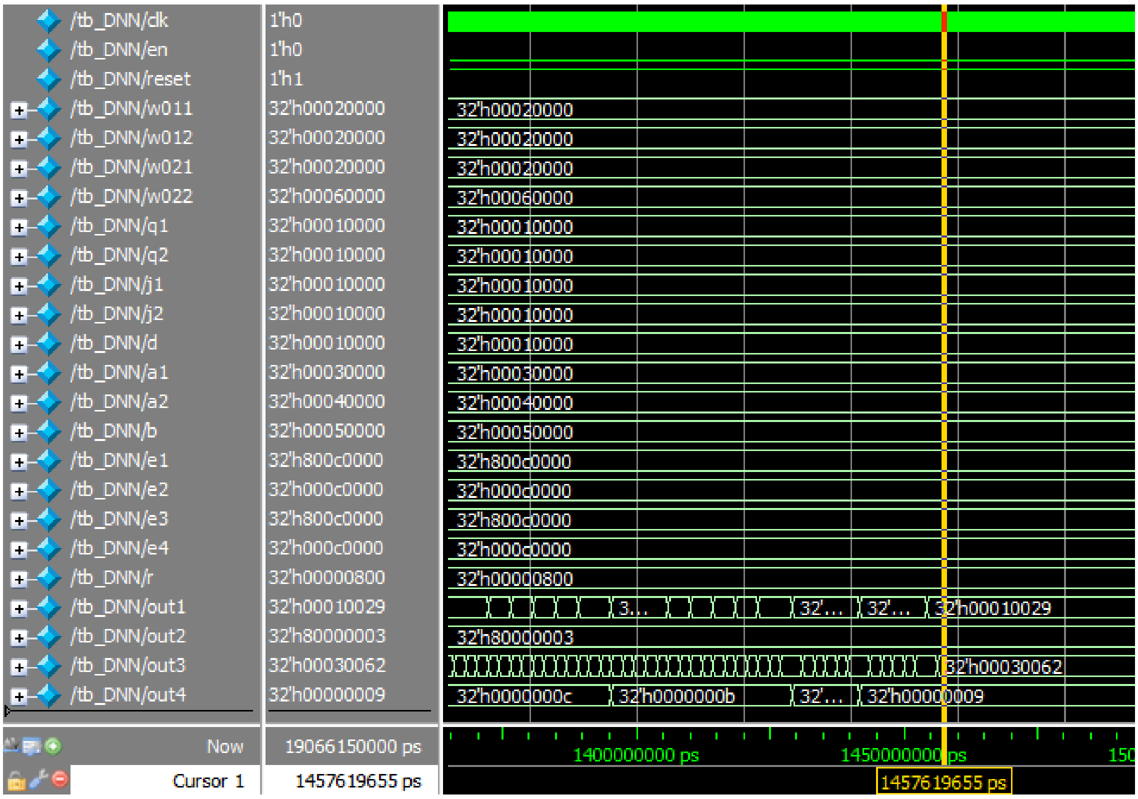
Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

x =

```
1.0000
0.0000
```

比较DNN模块在modelsim下的仿真结果：



由图：$x_1 = out1 = 32'h0001\_0029 \approx 1.0006$

$x_2 = out2 = 32'h8000\_0003 \approx -0.00004$

结论：仿真结果与matlab计算结果相比，很好的证明了本次网络设计的可行性和准确性。

四、复现工作中遇到的一些问题和解决方法（Q&A）

Q1：激活函数模块（function1）仿真结果有误

A1：尝试用流水的方法将其重写，仿真并验证得正确结果

Q2：四输入加法器模块（note_acc）modelsim仿真过程会出现warning，即仿真初始逻辑状态不定的情况

A2：每个模块的输入口输出口都要定义一个初始状态

Q3：存储模块（memory）仿真末端出现fatal error

A3：是由于tb文件中测试地址码数据超过7位数组导致，整体模块实现中，地址模块（addr_gen）输出地址码控制在7位数据以内，故可以忽略此error

五、还有一些问题尚未解决可以着手改善（Q）

Q1：基于定点数的神经网络的FPGA实现，运算精度会受影响，考虑基于浮点数的神经网络，需改变加法器等模块的计算方法。

Q2：该论文仅进行了功能仿真，还需要考虑器件延时以及布线延时，即需要进一步进行时序仿真。这点可以在addr_gen以及state_ctrl模块中增长每个步骤之间的时延。

Q3：时序仿真之后即可将代码烧录入板子中。可考虑利用单片机将初始状态的数据并行输入FPGA的I/O口中。

Q4：目前市面上的FPGA上存储资源十分有限，故对于多维神经网络基础参数的存储

六、阅读的一些文献和博客

[1]Zhang Y , Li X , Zhang Z , et al. An LVI-based numerical algorithm for solving quadratic programming problems, *Oper. Res. Trans.* 16 (2012), no. 1, 21–30.

[2]Zhang Y , Ma W , Li X , et al. MATLAB Simulink modeling and simulation of LVI-based primal–dual neural network for solving linear and quadratic programs, Neurocomputing, Volume 72, Issues 7–9, 2009, Pages 1679-1687, ISSN 0925-2312,

[3]Zhang Y . On the LVI-based primal-dual neural network for solving online linear and quadratic programming problems, American Control Conference, 2005. Proceedings of the 2005. IEEE, 2005.

[4]Zhang Y , Cai B , Zhang L , et al. Bi-criteria Velocity Minimization of Robot Manipulators Using a Linear Variational Inequalities-Based Primal-Dual Neural Network and PUMA560 Example, Advanced Robotics, 2008, 22(13-14):1479-1496.

[5]https://github.com/ljpzzz/machinelearning

[6]https://github.com/josephmisiti/awesome-machine-learning

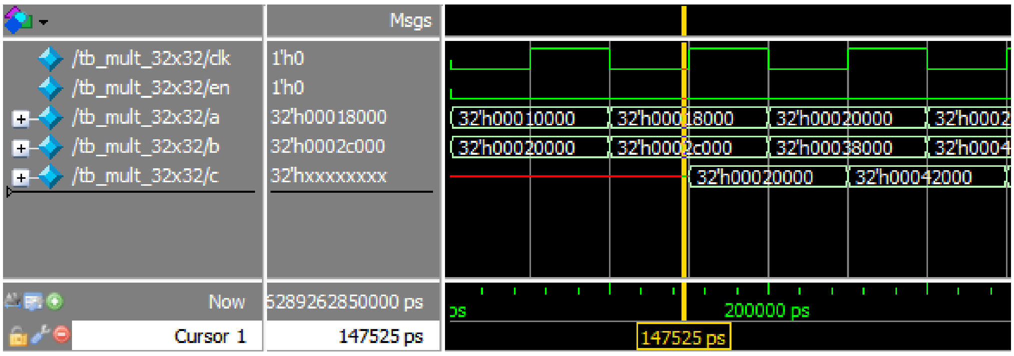
[7]https://blog.csdn.net/cxk207017/article/details/90736697

[8]https://ww2.mathworks.cn/help/optim/ug/quadprog.html?requestedDomain=cn

附录：各子模块的仿真与波形分析

原论文在验证部分仅贴出波形图，并未对波形数据进行说明举证，这里做一些补充。

1.乘法器模块（mult_32x32）

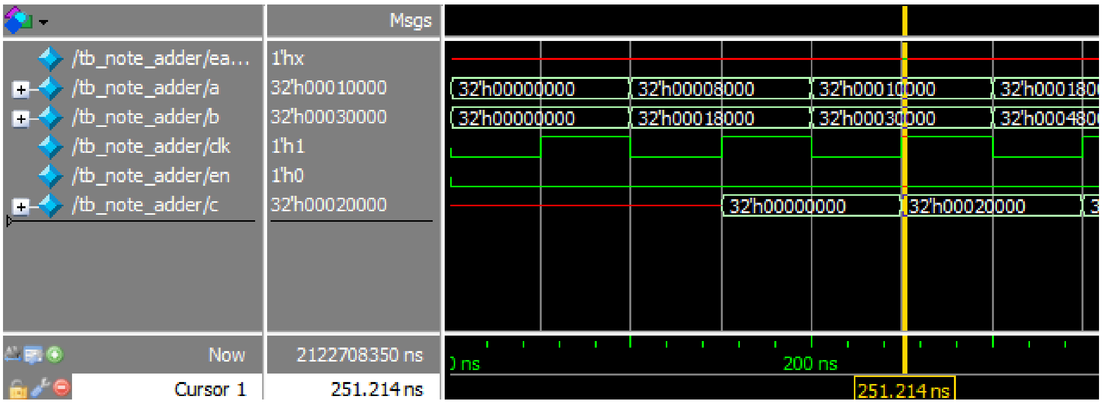


输入参数：$a=32'h0001\_0000=1; b=32'h0002\_0000=2$

一周期后：$c=32'h0002\_0000=2=1\times 2=a\times b$

即该模块验证完毕。

2.加法器模块（note_adder）



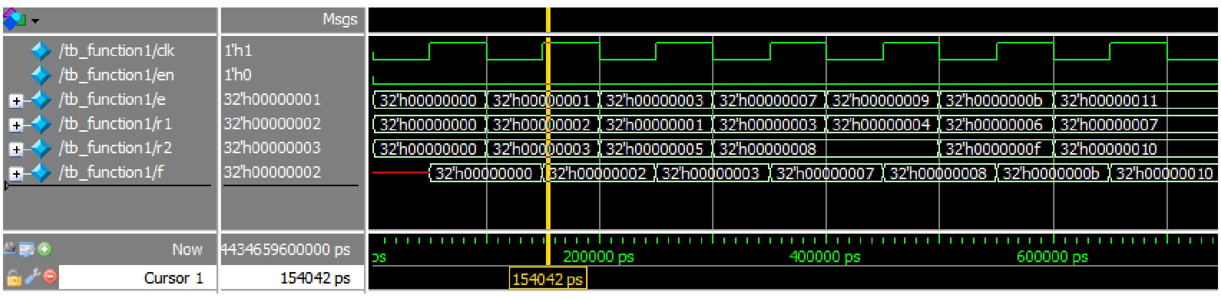
输入参数：$a=32'h0000\_8000=0.5; b=32'h0001\_8000=1.5$

一周期后：$c=32'h0002\_0000=2=0.5+1.5=a+b$

即该模块验证完毕。

3.激活函数模块

a.function1



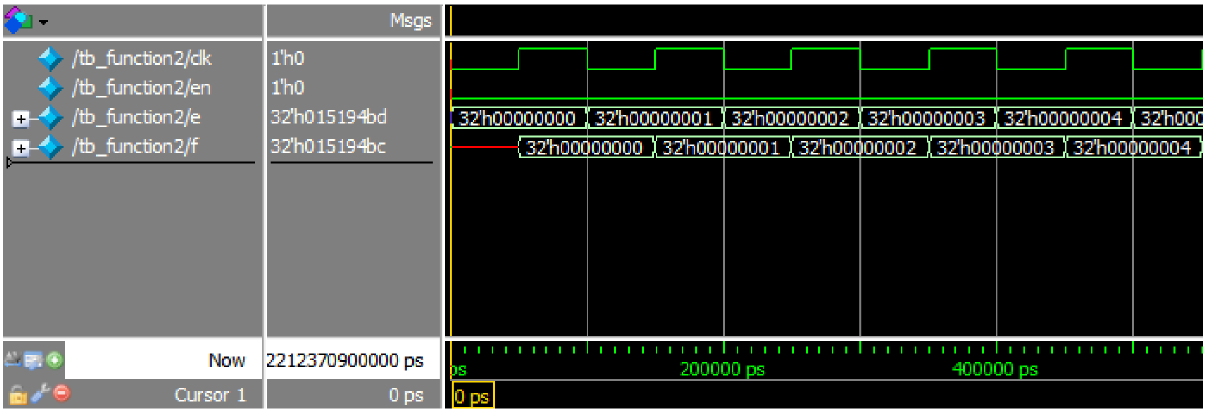
输入参数：$e=[0;1;3;7;9;11;17]$

边界参数：$r_1 = [0;2;1;3;4;6;7]; r_2 = [0;3;5;8;8;15;16]$

输出参数：$f = [0;2;3;7;8;11;16]$

即验证该function1函数模块的特性曲线为：

b.fuction2

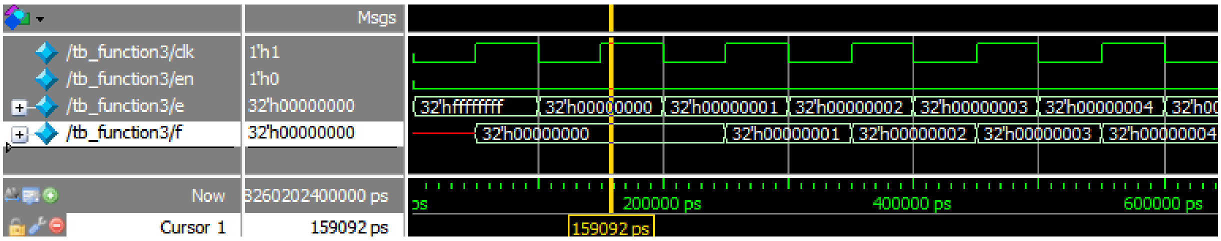


输入参数：$e = [0;1;2;3;4]$

输出参数：$f = [0;1;2;3;4]$

即验证该function2函数模块的特性曲线为：

c.function3

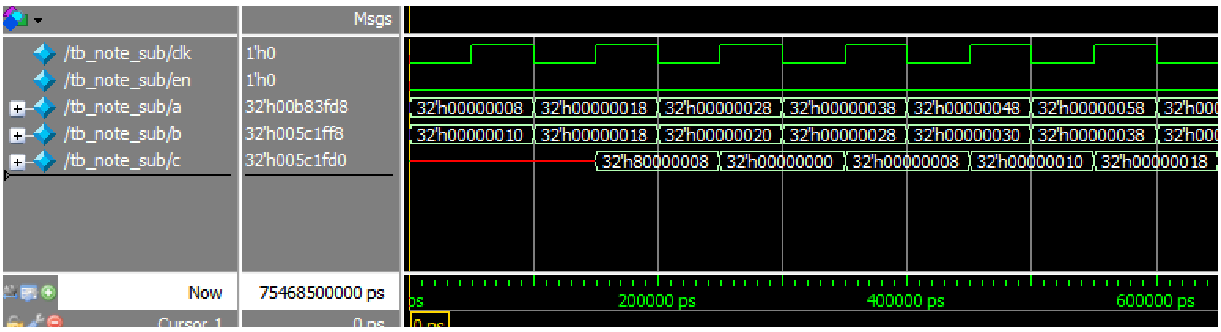


输入参数：$e = [-1;0;1;2;3;4]$

输出参数：$f = [0;0;1;2;3;4]$

即验证该function3函数模块的特性曲线为：

4.减法器模块（note_sub）

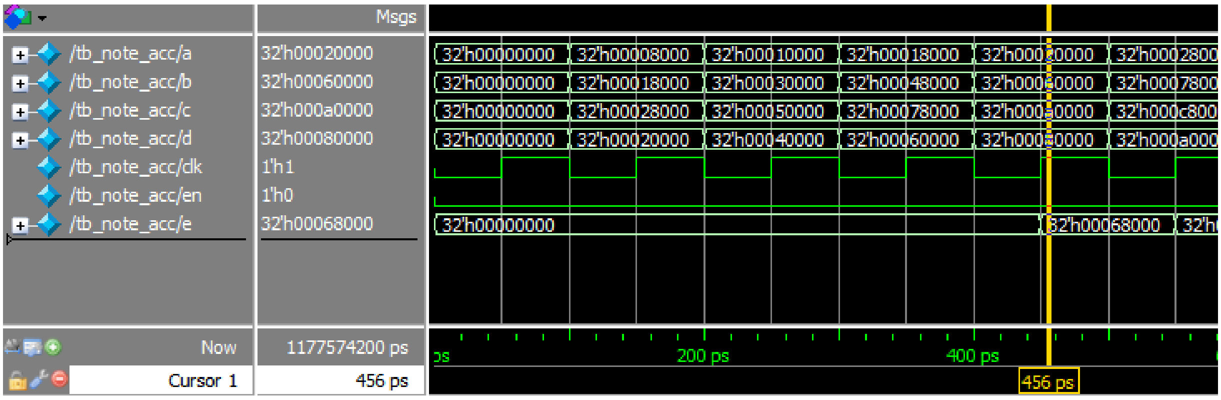


输入参数：$a = 32'h0000\_0008 = 32'd8; b = 32'h0000\_0010 = 32'd16$

输出参数：$c = 32'h8000\_0008 = 32'd-8 = 32'd8 - 32'd16 = a - b$

即该模块验证完毕

5.四输入累加器模块（note_acc）



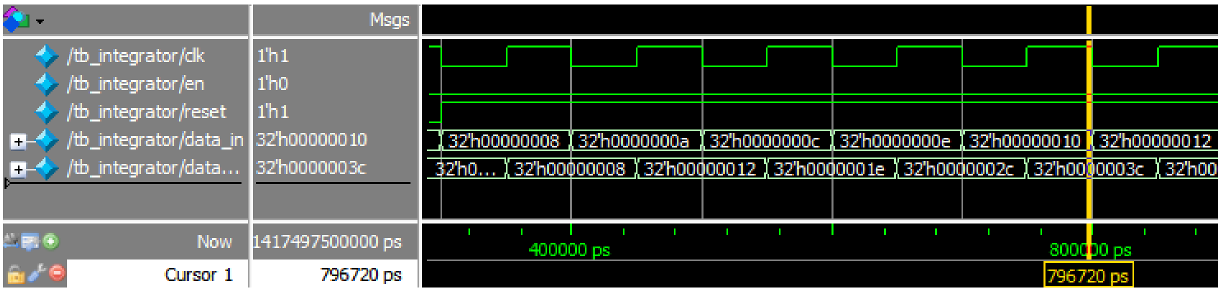
输入参数：$a = 32'h0000\_8000 = 0.5; b = 32'h0001\_8000 = 1.5$

$c = 32'h0002\_8000 = 2.5; d = 32'h0002\_0000 = 2$

输出参数：$e = 32'h0006\_8000 = 6.5$

即该模块验证完毕

6.积分器模块（Integrator）

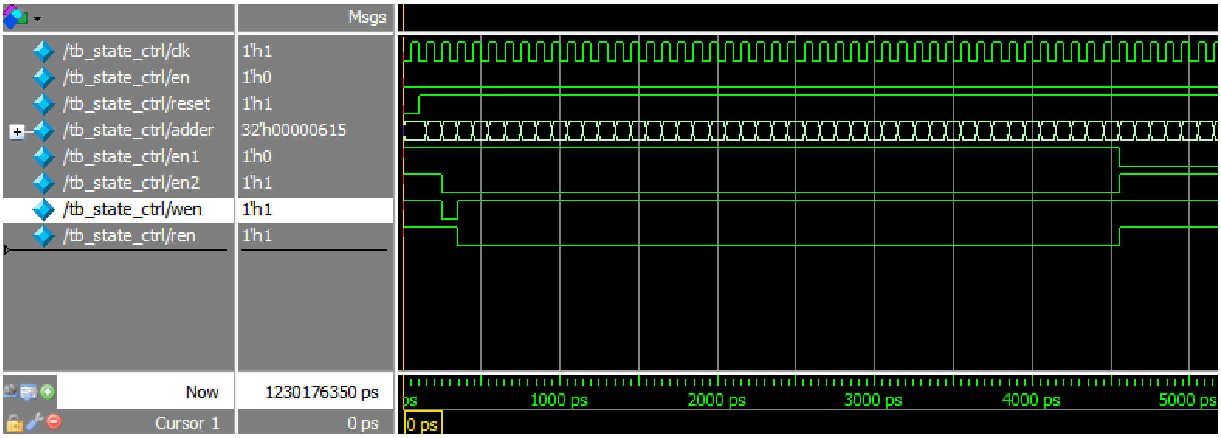


输入参数：$data\_in = [8, 10, 12, 14, 16, 18]$

输出参数：$data\_out = [8, 18, 30, 44, 60, 78]$

即该模块验证完毕

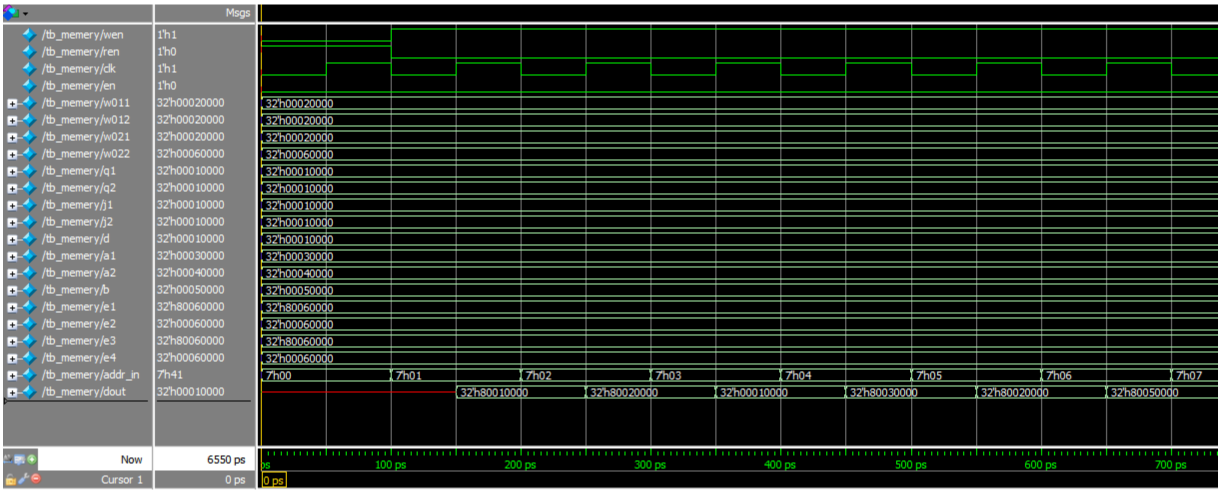
7.状态控制模块（state_ctrl）



由图：（en1,en2,wen,ren）第一个周期为4'b1111，第三个周期为4'b1001，第四个周期为4'b1010，第46个周期为4'b0111，即实现了初始状态到写状态到读状态再到计算状态的转换。

即该模块验证完毕

8.存储模块（memory）



输入地址码$addr_in$由0增加到39，即实现了dout依次输出40个基础参数。

将实例中的参数输入模块中，可得：
$dout = [-1;-2;1;-3;-2;-5;1;-4;-1;-1;1;0;3;4;0;1;1;1;-1;5;-6;6;-6;6;3;2;1;-3;2;7;1;-4;-1;-1;1;0;3;4;0;1]$

可手算验证这些参数的正确性。