模块指令集

目录

[1.指令综述 2](#_Toc59442272)

[2.各模块介绍 3](#_Toc59442273)

[BNN\_Core 3](#_Toc59442274)

[BPU\_Group 5](#_Toc59442275)

[BPU 6](#_Toc59442276)

1.指令综述：

|  |  |  |
| --- | --- | --- |
| 信号 | **具体指令及功能** 按inst[13:0]格式 | 功能 |
| Instruction 14bits  Bnn\_Core指令 | **Instruction\_to\_bpug** | [7:0]：Inst\_to\_bpug  给BPUG的指令  Core的MUX信号复用前八位中的lut\_sel  Core中PSUM\_RST也复用[0] |
| **PSUM进行一次累加：**  Inst[9]=1，[4:1]=某个数, others=0  (0≤某个数≤4’b1111)  Next posedge inst[9]=0 | [9]：bpug\_psum\_add:  Psum进行一次累加  [4:1]bpug\_sel：  选出需要做累加的存放BPUG计算结果的寄存器 |
| **将二值化计算结果写入某个寄存器：**  [10]=1  Next posedge [10] = 0 | [10]：Cal\_bin\_wr 1bit  将二值化计算结果写入REG中  [12]：是否pooling  [6][13]：写入pooling reg哪一位 |
| **写入BIAS：**  [11] = 1, others = 0  2 posedges later [11]=0 | [11]：Bias\_wr 1bit  BIAS REG写入使能 |
| **PSUM\_REG赋为BIAS**  [0]=1  Next posedge [0]=0 | [0]：psum\_rst  **此条指令是不必要的**，它会与**BPUG中PSUM置0**同时发生。有紫色指令即可。 |
| BPUG指令  8bits | **Instruction\_to\_bpu** | Inst\_to\_bpu 5bit  给BPU的指令 |
| **从8列img\_reg中选7列**  [6]=0 or 1 | [6]：Data\_sel  选通INGPUT\_REG的某七列  它只在计算期间起作用 |
| **写入img或wgt数据**  [8] or [7] =1, [2:1]=某个数others = 0  （0≤某个数≤2’b11）  Next posedge all = 0 | [8:7]：EN 2bit  IMG和WGT REG的使能信号  [2:1]：bpug\_sel  写入时的Bpug选通信号 |
| BPU指令  5bits | **PSUM寄存器置零**  [0]=1, others = 0 | [0]：Psum\_rst 1bit  置高时，PSUM置零 |
| **算某个BPUE同或结果的popcnt**  [3:1] ≤3’b111 | [3:1]：Lut\_sel 3bits  选通某个BPUE结果进入LUT |
| **Popcnt值进行累加**  [5]=1,[3:1]不变，其余=0 | [5]：Psum\_add 1bit  LUT结果进入PSUM累加 |

# 2.各模块介绍

## BNN\_Core



功能：通过MUX逐个选出BPUG的计算结果并进行求和，求和结果存于PSUM中。求和结果可选做2x2 Pooling。

输入信号：

|  |  |  |
| --- | --- | --- |
| 信号 | 具体 按inst[13:0]格式 | 功能 |
| Instruction 14bits  指令 | Depends | [7:0]：Inst\_to\_bpug  给BPUG的指令  Core的MUX信号复用前八位中的lut\_sel  Core中PSUM\_RST也复用[0] |
| **PSUM进行一次累加：**  Inst[9]=1，[4:1]=某个数, others=0  (0≤某个数≤4’b1111)  Next posedge inst[9]=0 | [9]：bpug\_psum\_add:  Psum进行一次累加  [4:1]bpug\_sel：  选出需要做累加的存放BPUG计算结果的寄存器 |
| **将二值化计算结果写入某个寄存器：**  [10]=1  Next posedge [10] = 0 | [10]：Cal\_bin\_wr 1bit  将二值化计算结果写入输出REG中  [12]：是否pooling  [6][13]：一起判断写入pooling reg的位置 |
| **写入BIAS：**  [11] = 1, others = 0  2 posedges later [11]=0 | [11]：Bias\_wr 1bit  BIAS REG写入使能 |
| **PSUM\_REG赋为BIAS**  [0]=1  Next posedge [0]=0 | [0]：psum\_rst  **此条指令是不必要的**，它会与**BPUG中PSUM置0**同时发生。有紫色指令即可。 |
| clk | - | 时钟信号 |
| rst | - | 复位信号，寄存器全部置零 |
| Data\_in 4×8bits | 32bits数据总线 | EN信号控制写入哪个REG |

输出信号

|  |  |  |
| --- | --- | --- |
| 信号 | 具体 | 功能 |
| Reg\_bins 8bits | 8个二值化结果 | 每个BPU的计算结果 |

寄存器：

|  |  |
| --- | --- |
| 名称 | 作用 |
| Pooling\_reg | 存放pooling中间值 |
| Bias 8×7bits | 存放bias数值 |
| Cal\_intern 8×9bits | 存放全精度计算结果，它就是框图中的PSUM |
| Reg\_bins | 存放本轮计算输出结果 |

## BPU\_Group



功能：将图像及权重存入两组REG中并进行计算，结果通过OUT输出。

输入信号：

|  |  |  |
| --- | --- | --- |
| 信号 | 具体 按inst[13:0]格式 | 功能 |
| Instruction 8bits  指令 | Depends | Inst\_to\_bpu 5bit  给BPU的指令 |
| **从8列img\_reg中选7列**  [6]=0 or 1 | [6]：Data\_sel  选通INGPUT\_REG的某七列  它只在计算期间起作用 |
| **写入img或wgt数据**  [8] or [7] =1, [3:1]=某个数others = 0  （0≤某个数≤2’b11）  Next posedge all = 0 | [8:7]：EN 2bit  IMG和WGT REG的使能信号  [3:1]：bpug\_sel  Bpug选通信号 |
| clk | - | 时钟信号 |
| rst | - | 复位信号，寄存器全部置零 |
| Data\_in 8bits | 连到IMG和WGT REG | EN信号控制写入哪个REG |

输出信号

|  |  |  |
| --- | --- | --- |
| 信号 | 具体 | 功能 |
| Bpu\_out 6×7bits | 6×7bits 有符号数 | 每个BPU的计算结果 |

寄存器

|  |  |
| --- | --- |
| 名称 | 作用 |
| wgt | 存放权重数据 |
| Img\_reg | 存放图像数值 |

## BPU



功能：BPUE对图像和权重的一列数据进行同或计算，结果逐列通过MUX进入查找表进行popcount计算。 BPUE进行1×7bits的同或计算。

输入信号：

|  |  |  |
| --- | --- | --- |
| 信号 | 具体 按inst[13:0]格式 | 功能 |
| Instruction 5bits  指令 | **PSUM寄存器置零**  [0]=1, others = 0 | [0]：Psum\_rst 1bit  置高时，PSUM置零 |
| **算某个BPUE同或结果的popcnt**  [3:1]={0,1,2,3,4,5,6,7} | [3:1]：Lut\_sel 3bits  选通某个BPUE结果进入LUT |
| **Popcnt值进行累加**  [5]=1,[3:1]不变，其余=0 | [5]：Psum\_add 1bit  LUT结果进入PSUM累加 |
| clk | - | 时钟信号 |
| rst | - | 复位信号，寄存器全部置零 |
| Img 7×7bits | 每7bits接入一个BPUE | 计算的图像数据 |
| Wgt 7×7bits | 每7bits接入一个BPUE | 计算的权重数据 |

输出信号

|  |  |  |
| --- | --- | --- |
| 信号 | 具体 | 功能 |
| Popcnt\_add 7bits | 7bits 有符号数 | POPCOUNT计算结果 |

寄存器

|  |  |
| --- | --- |
| 名称 | 作用 |
| popcnt | 存放1×7popcount计算结果 |
| Popcount\_add | 存放7×7popcount计算结果 |