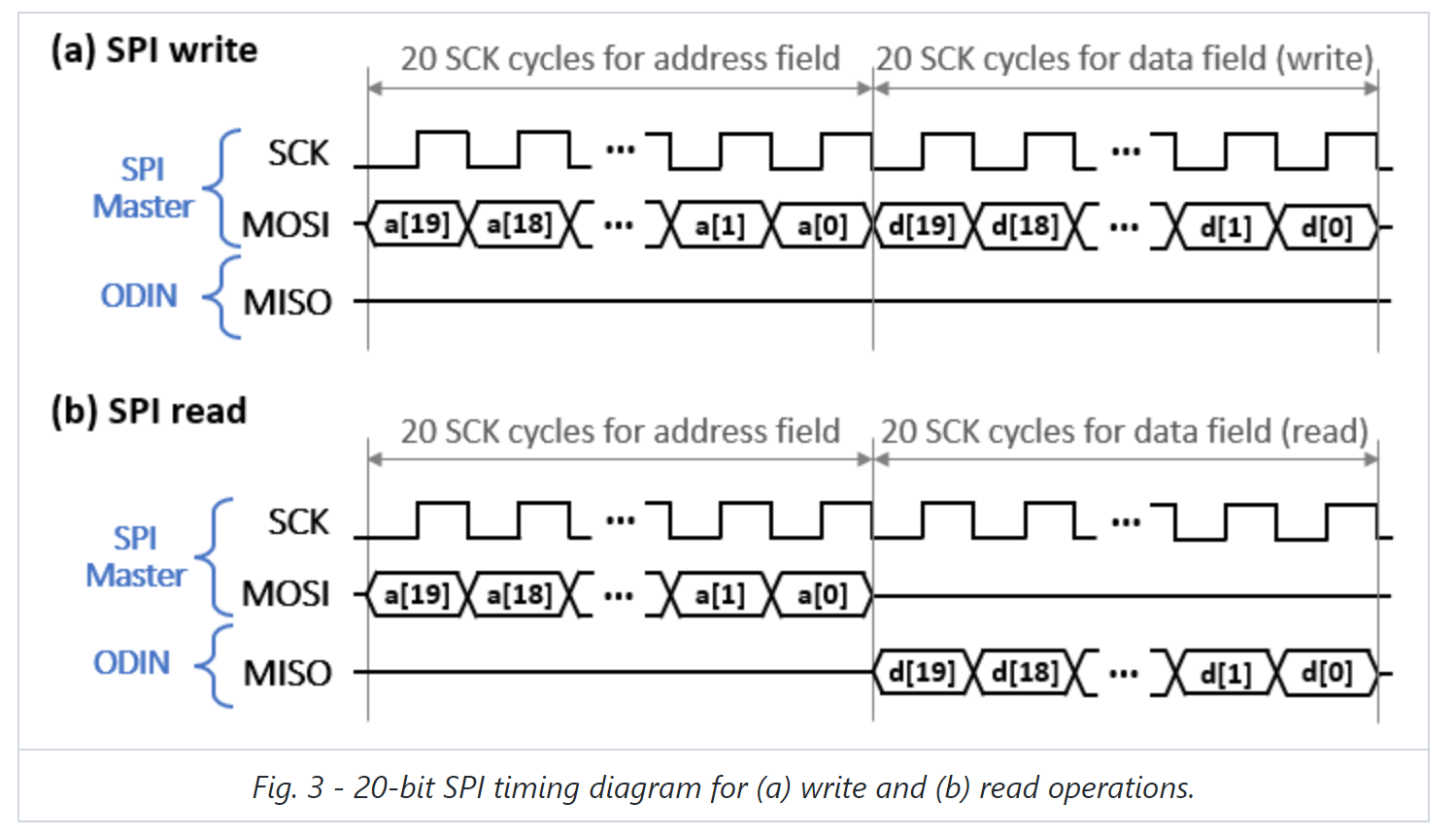
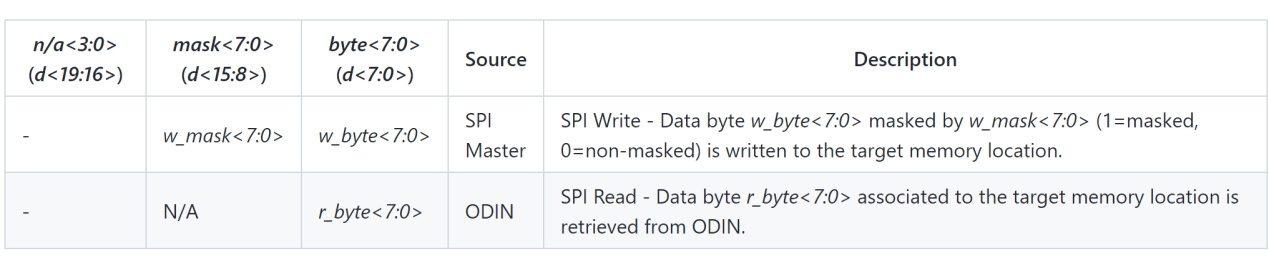
**一、SPI配置寄存器——对神经元和突触进行编写和回读**

****

**图形用户界面, 文本, 应用程序

描述已自动生成**

****

**Spi-write时：**

**见上表：**

**当addr[19:0] 高四位为0，则对26个寄存器进行配置；**

**当addr[19:0] 高四位不为0，则对神经元或者突触进行读写操作。**

**1、写神经元 SPI配置256个神经元 每个神经元128bit**

**If (i == CTRL\_SPI\_ADDR[15:8])**

**assign neuron\_data[8i+7:8i] =**

**((CTRL\_PROG\_DATA[7:0] & ~CTRL\_PROG\_DATA[15:8]) | (NEUR\_STATE[8i+7:8i] & CTRL\_PROG\_DATA[15:8]))**

CTRL\_PROG\_DATA = 00000000 01001011 表示神经元数据来自**CTRL\_PROG\_DATA**低8位

CTRL\_PROG\_DATA = 11111111 01001011 表示神经元数据来自**NEUR\_STATE[8i+7:8i]**的8位

因此如果是写神经元那么，应该使得CTRL\_PROG\_DATA = 00000000 01001011

即CTRL\_PROG\_DATA =00000000 \*\*\*\*\*\*\*\*时： **neuron\_data[8i+7:8i] =CTRL\_PROG\_DATA[7:0]**

**spi配置一个神经元的128bit的参数：{0101 CTRL\_SPI\_ADDR[15:8]，0000 CTRL\_PROG\_DATA[15:0]}**

|  |  |
| --- | --- |
| **spi\_shift\_reg\_in[19:0]** | **spi\_shift\_reg\_in[39:20]** |
| **0101 CTRL\_SPI\_ADDR[15:0]=** | **0000** **CTRL\_PROG\_DATA[15:0]= 下面每一行的低8位组合依次是神经元的128个比特的地位到高位** |
| **0101** 00000000 0000000**0 [n0]**  **0101** 00000001 0000000**0**  **0101** 00000010 0000000**0**  **0101** 00000011 0000000**0**  **0101** 00000100 0000000**0**  **0101** 00000101 0000000**0**  **0101** 00000110 0000000**0**  **0101** 00000111 0000000**0**  **0101** 00001000 0000000**0**  **0101** 00001001 0000000**0**  **0101** 00001010 0000000**0**  **0101** 00001011 0000000**0**  **0101** 00001100 0000000**0**  **0101** 00001101 0000000**0**  **0101** 00001110 0000000**0**  **0101** 00001111 0000000**0** | **\*\*\*\*** 00000000 00000101  **\*\*\*\*** 00000000 01000001  **\*\*\*\*** 00000000 00010011  **\*\*\*\*** 00000000 01000000  **\*\*\*\*** 00000000 00010100  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000 |
| **0101** 00000000 0000000**1 [n1]**  **0101** 00000001 0000000**1**  **0101** 00000010 0000000**1**  **0101** 00000011 0000000**1**  **0101** 00000100 0000000**1**  **0101** 00000101 0000000**1**  **0101** 00000110 0000000**1**  **0101** 00000111 0000000**1**  **0101** 00001000 0000000**1**  **0101** 00001001 0000000**1**  **0101** 00001010 0000000**1**  **0101** 00001011 0000000**1**  **0101** 00001100 0000000**1**  **0101** 00001101 0000000**1**  **0101** 00001110 0000000**1**  **0101** 00001111 0000000**1** | **\*\*\*\*** 00000000 00000101  **\*\*\*\*** 00000000 01000001  **\*\*\*\*** 00000000 00010011  **\*\*\*\*** 00000000 01000000  **\*\*\*\*** 00000000 00010100  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000 |
| **………………** | **………………** |
| **0101** 00000000 **11111111 [n255]**  **0101** 00000001 **11111111**  **0101** 00000010 **11111111**  **0101** 00000011 **11111111**  **0101** 00000100 **11111111**  **0101** 00000101 **11111111**  **0101** 00000110 **11111111**  **0101** 00000111 **11111111**  **0101** 00001000 **11111111**  **0101** 00001001 **11111111**  **0101** 00001010 **11111111**  **0101** 00001011 **11111111**  **0101** 00001100 **11111111**  **0101** 00001101 **11111111**  **0101** 00001110 **11111111**  **0101** 00001111 **11111111** | **\*\*\*\*** 00000000 00000101  **\*\*\*\*** 00000000 01000001  **\*\*\*\*** 00000000 00010011  **\*\*\*\*** 00000000 01000000  **\*\*\*\*** 00000000 00010100  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000  **\*\*\*\*** 00000000 00000000 |

配置完神经元要退出W\_NEUR的状态，

如果配置突触权重，会自动退出写神经元的状态。

同理突触权重配写完之后，也需要退出写突触权重的状态。

如果自动配置之前写好的26个寄存器，那系统就会跳出配置状态，进入正常的工作。

**2、写突触权重 SPI配置（256\*32行突触权重 每个突触权重4-bit）**

|  |  |
| --- | --- |
| **Spi-addr[19:0]**  **spi\_shift\_reg\_in <= {spi\_shift\_reg\_in[18:0], MOSI};** | **Spi\_data[19:0]** |
| **0110 \*00 0000000000000 第1行第1格的2个权重**  **0110 \*01 0000000000000 第1行第2格的2个权重**  **0110 \*10 0000000000000 第1行第3格的2个权重**  **0110 \*11 0000000000000 第1行第4格的2个权重**  **0110 \*00 0000000000001 第2行第1格的2个权重**  **0110 \*01 0000000000001 第2行第2格的2个权重**  **0110 \*10 0000000000001 第2行第3格的2个权重**  **0110 \*11 0000000000001 第2行第4格的2个权重**  **…………………………**  **0110 \*00 1111111111111 第256\*32行第1格的2个权重**  **0110 \*01 1111111111111 第256\*32行第2格的2个权重**  **0110 \*10 1111111111111 第256\*32行第3格的2个权重**  **0110 \*11 1111111111111 第256\*32行第4格的2个权重** | **0000000000000 w0\_00[7:0]**  **0000000000000 w0\_01[7:0]**  **0000000000000 w0\_10[7:0]**  **0000000000000 w0\_11[7:0]**  **0000000000000 w1\_00[7:0]**  **0000000000000 w1\_01[7:0]**  **0000000000000 w1\_10[7:0]**  **0000000000000 w1\_11[7:0]**  **………………**  **0000000000000 w256\*32-1\_00[7:0]**  **0000000000000 w256\*32-1\_\_01[7:0]**  **0000000000000 w256\*32-1\_\_10[7:0]**  **0000000000000 w256\*32-1\_\_11[7:0]** |

**3、读神经元 SPI配置（256\*16）16\*8=128bit**

**4、读突触权重 SPI配置（256\*32行突触权重 每个突触权重4-bit）**

|  |  |
| --- | --- |
| **spi\_shift\_reg\_in[19:0]** | **spi\_shift\_reg\_in[39:20]** |
| **1010 CTRL\_SPI\_ADDR[15:0]** | **0000000000000 w[7:0] 一次只读一个存储单元的权重（2个8-bit）** |
| **根据要读的突触权重的位置直接读取即可，即指定{n/a,*byte\_addr<1:0>*,*word\_addr<12:0>*}即可** | |
| **1010 \*00 0000000000001**  **读取第一行中右边第一格的2个突触权重** | **0000000000000 [01101011]**  **20bit只有低8bit：[01101011]，才是“00”byte\_addr对应地址的2个突触权重** |
| **如果要读取所有的突触权重：**  **从第一行开始到最后一行**  **1010 \*00 0000000000000 第1行第1格的2个权重**  **1010 \*01 0000000000000 第1行第2格的2个权重**  **1010 \*10 0000000000000 第1行第3格的2个权重**  **1010 \*11 0000000000000 第1行第4格的2个权重**  **1010 \*00 0000000000001 第2行第1格的2个权重**  **1010 \*01 0000000000001 第2行第2格的2个权重**  **1010 \*10 0000000000001 第2行第3格的2个权重**  **1010 \*11 0000000000001 第2行第4格的2个权重**  **…………………………**  **1010 \*00 1111111111111 第256\*32行第1格的2个权重**  **1010 \*01 1111111111111 第256\*32行第2格的2个权重**  **1010 \*10 1111111111111 第256\*32行第3格的2个权重**  **1010 \*11 1111111111111 第256\*32行第4格的2个权重**  **这么多行，存到txt文件中当作字符串读出来，给到ODIN;**  **同时还需要接收返回的突触权重** | **读到的权重也有20bit，只有低位的8bit才是权重**  **0000000000000 w0\_00[7:0]**  **0000000000000 w0\_01[7:0]**  **0000000000000 w0\_10[7:0]**  **0000000000000 w0\_11[7:0]**  **0000000000000 w1\_00[7:0]**  **0000000000000 w1\_01[7:0]**  **0000000000000 w1\_10[7:0]**  **0000000000000 w1\_11[7:0]**  **………………**  **0000000000000 w256\*32-1\_00[7:0]**  **0000000000000 w256\*32-1\_\_01[7:0]**  **0000000000000 w256\*32-1\_\_10[7:0]**  **0000000000000 w256\*32-1\_\_11[7:0]** |

**五、26个spi寄存器的修改**

**如果需要对神经元和突触进行编写和回读，修改SPI\_GATE\_ACTIVITY=1，即**

**通过spi输入 00000000000000000000 00000000000000000001**

**如果进行测试，需要关闭SDSP，修改SPI\_UPDATE\_UNMAPPED\_SYN=0，即**

**通过spi输入 00000000000000010111 00000000000000000000 （当时钟数位40，最后一个比特1代表MOSI）**

**计数器计数40个时钟，将40个bit从左到右全部读入，然后才开始赋值。**

|  |  |  |  |
| --- | --- | --- | --- |
| **寄存器** | **[19:0]** | **[19:0]** | **说明** |
| **SPI\_GATE\_ACTIVITY** | **00000000000000000000‘** | **00000000000000000000‘** | **第40bit为0，不对神经元和突触内存进行编程和回读** |
| **SPI\_OPEN\_LOOP** | **00000000000000000001‘** | **00000000000000000001‘** | **第40bit为1，不允许突触后神经元产生的脉冲进入调度器中。** |
| **SPI\_SYN\_SIGN[15：0]** | **00000000000000000010‘** | **00000000000000000000‘** | **将256个神经元中的每一个配置为抑制性或兴奋性**  **第40bit为0代表兴奋性神经元** |
| **SPI\_SYN\_SIGN[31：16]** | **00000000000000000011‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[47：32]** | **00000000000000000100‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[63：48]** | **00000000000000000101‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[79：64]** | **00000000000000000110‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[95：80]** | **00000000000000000111‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[111：96]** | **00000000000000001000‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[127：112]** | **00000000000000001001‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[143：128]** | **00000000000000001010‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[159：144]** | **00000000000000001011‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[175：160]** | **00000000000000001100‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[191：176]** | **00000000000000001101‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[207：192]** | **00000000000000001110‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[223：208]** | **00000000000000001111‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[239：224]** | **00000000000000010000‘** | **00000000000000000000‘** |
| **SPI\_SYN\_SIGN[255：240]** | **00000000000000010001‘** | **00000000000000000000‘** |
| **SPI\_BURST\_TIMEREF** | **00000000000000010010‘** | **00000000000000000000‘** | **第40bit为0，ODIN输出的事件直接来自突触后神经元激发的脉冲** |
| **SPI\_AER\_SRC\_CTRL\_nNEUR** | **00000000000000010011‘** | **00000000000000000000‘** |  |
| **SPI\_OUT\_AER\_MONITOR\_EN** | **00000000000000010100‘** | **00000000000000000000‘** |  |
| **SPI\_MONITOR\_NEUR\_ADDR** | **00000000000000010101‘** | **00000000000000000000‘** |  |
| **SPI\_MONITOR\_SYN\_ADDR** | **00000000000000010110‘** | **00000000000000000000‘** |  |
| **SPI\_UPDATE\_UNMAPPED\_SYN** | **00000000000000010111‘** | **00000000000000000001‘** | **第40bit为1允许突触权重进行SDSP在线学习** |
| **SPI\_PROPAGATE\_UNMAPPED\_SYN** | **00000000000000011000‘** | **00000000000000000001‘** | **第40bit为0允突触权重可以作用于突触后神经元** |
| **SPI\_SDSP\_ON\_SYN\_STIM** | **00000000000000011001‘** | **00000000000000000001‘** | **第40bit为0对于synapse\_event，使能SDSP在线学习** |