

目录

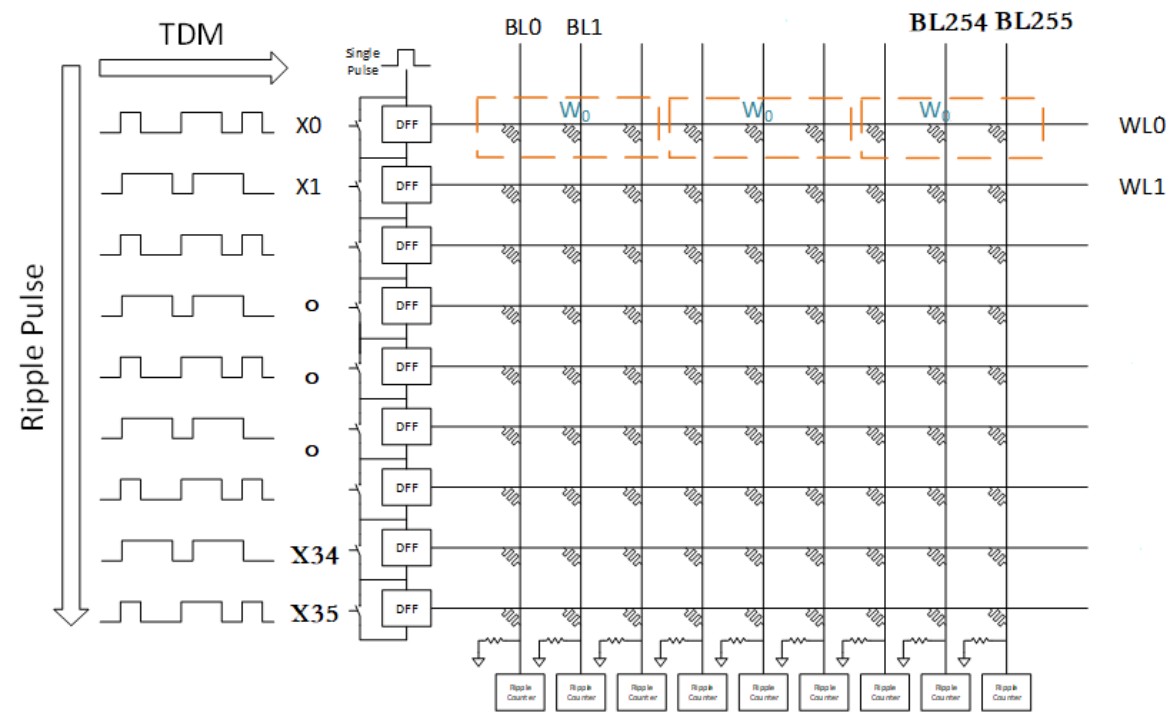
- 1. 设计需求
- 2. 核心架构
- 3. 端口
  - 3.1 端口列表
  - 3.2 端口时序
    - 3.2.1 运算阶段端口时序
    - 3.2.1 写RRAM阶段端口时序
    - 3.2.1 读RRAM阶段端口时序

1. 设计需求

设计基于RRAM阵列的计算核心，实现：

- $Y = W^T X$ ，其中 $X$ 为 $36 \times 1$ 的输入数据向量， $W$ 为 $36 \times 32$ 的权值矩阵， $Y$ 为 $32 \times 1$ 的输出向量。数据格式为有符号int8
- $36 \times 32 \times 8$  RRAM单元的基本读写功能

2. 核心架构



3. 端口

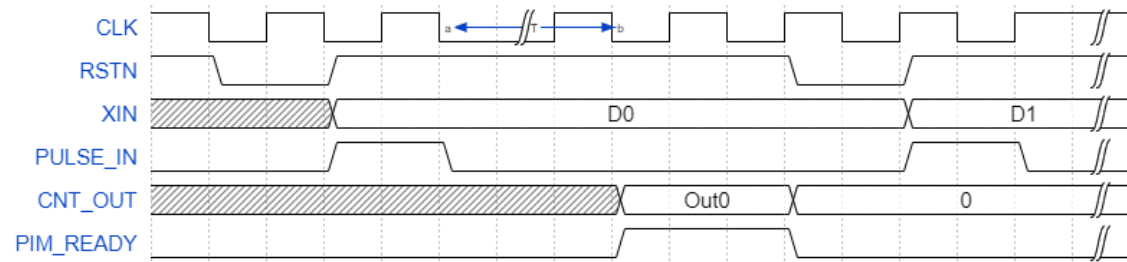
3.1 端口列表

name	type	width	description
CLK	input	1	时钟信号
RSTN	input	1	复位信号，低有效
XIN	input	36	数据输入
PULSE_IN	input	1	数据有效信号
CNT_OUT	output	256	每bit代表其中一列的部分和脉冲（满足行波计数器需求）
PIM_READY	output	1	计数器输出有效标志

name	type	width	description
BL_ADDRESS	input	8	阵列位线选择地址；256列
BL_EN	input	1	位线地址信号READY
BL_WORK_MODE	input	1	位线工作模式；0：写RRAM；1：PIM
WL_ADDRESS	input	6	阵列字线选择地址；36行
WL_EN	input	1	字线地址信号READY
WL_WORK_MODE	input	1	字线工作模式；0：写RRAM；1：PIM
RRAM_SET	input	1	写RRAM，置1，高有效
RRAM_RSET	input	1	写RRAM，置0，高有效

## 3.2 端口时序

### 3.2.1 运算阶段端口时序



如上图所示，存算单元在正常运算阶段，在每一次运算之前，需要进行一次复位；在输入数据(XIN)准备好之后，将PULSE\_IN拉高一个时钟，在下次PULSE\_IN有效之前XIN必须保持不变。输出结果(CNT\_OUT)在PULSE\_IN使能之后 $T$ 个时钟周期有效，其中 $T$ 为XIN中bit为1的数量；数字部分在PIM\_READY信号为高的时候进行输出结果采样，复位时PIM\_READY信号为低电平。

XIN为36个数据以定义数据位宽(WID=8bit)分时输入，存算单元单次完成 $Y_p = W^T X_p$ ，其中 $X_p$ 为 $36 \times 1$ 的输入数据的某1bit， $W$ 为 $36 \times 32$ 的权值矩阵， $Y_p$ 为 $32 \times 1$ 的输出向量的部分和。

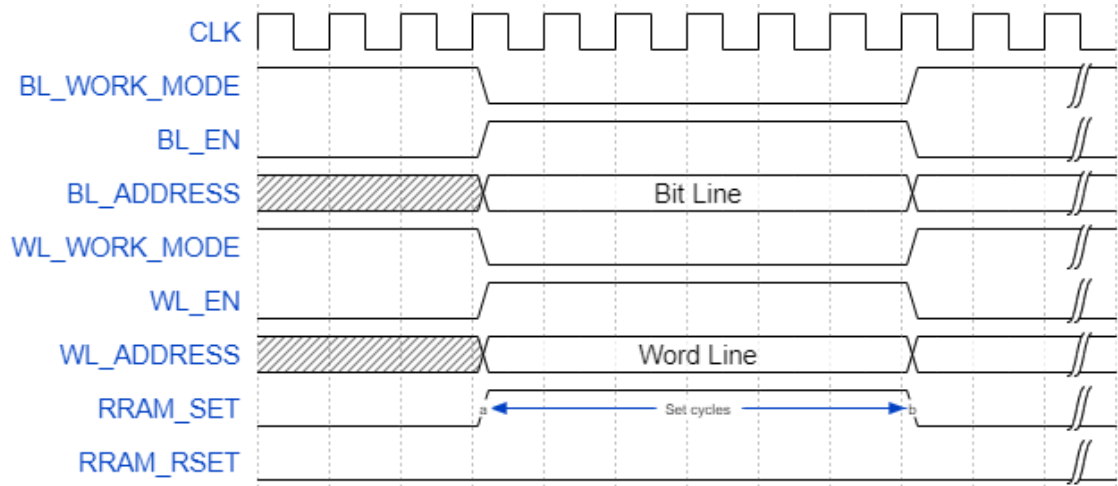
CNT\_OUT为256列运算结果脉冲输出（满足行波计数器，递增计数），每列为单bit脉冲形式输出供外部的行波计数器进行部分和结果计数（例如某列的部分和为N，则在该列上输出N个脉冲）。存算单元单次运算之后将每列部分和按照权值数据位宽按权进行累加；存算单元每次运算之间按输入数据位宽按权进行累加。

例:  $Result = X_{n-1}X_{n-2}\dots X_0 \times W_{n-1}W_{n-2}\dots W_0$ ,  $n$ 为数据位宽,  $X$ 为输入数据,  $W$ 为权重数据:

- 第1次运算:  $S_0 = X_0 * [(\sum_{i=0}^{n-2} W_i * 2^i) - W_{n-1} * 2^{n-1}]$
- 第2次运算:  $S_1 = X_1 * [(\sum_{i=0}^{n-2} W_i * 2^i) - W_{n-1} * 2^{n-1}]$
- 第n-1次运算:  $S_{n-1} = X_{n-1} * [(\sum_{i=0}^{n-2} W_i * 2^i) - W_{n-1} * 2^{n-1}]$
- $Result = (\sum_{j=0}^{n-2} S_j * 2^j) - S_{n-1} * 2^{n-1}$

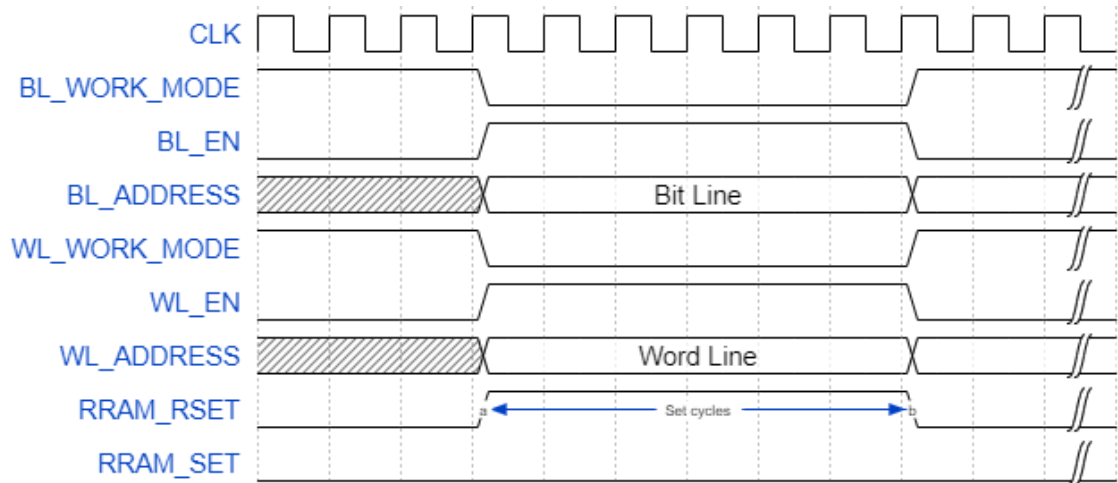
### 3.2.1 写RRAM阶段端口时序

- RRAM单元写1端口时序:



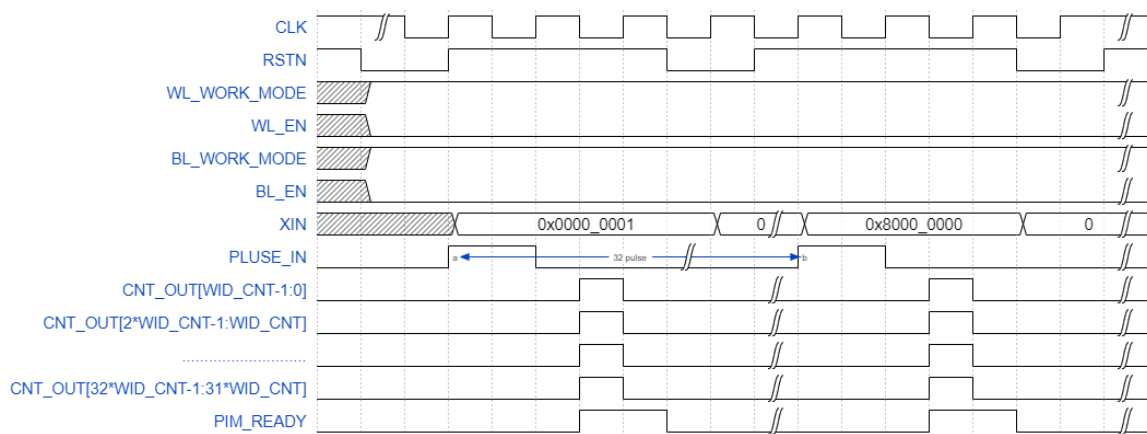
RRAM单元SET操作时, 将BL\_WORK\_MODE、BL\_EN、WL\_WORK\_MODE、WL\_EN、RRAM\_SET信号拉高, 同时通过字线地址WL\_ADDRESS、位线地址BL\_ADDRESS选择的相应单元, 经过Set cycles个时钟周期完成对一个RRAM单元的SET操作。

- RRAM单元写0端口时序:



RRAM单元RSET操作时, 将BL\_WORK\_MODE、BL\_EN、WL\_WORK\_MODE、WL\_EN、RRAM\_RSET信号拉高, 同时通过字线地址WL\_ADDRESS、位线地址BL\_ADDRESS选择的相应单元, 经过Set cycles个时钟周期完成对一个RRAM单元的RSET操作。

### 3.2.1 读RRAM阶段端口时序



读RRAM阶段，依次将**XIN**每bit置位1（one-hot编码），之后产生维持一个时钟周期的**PLUSE\_IN**脉冲信号；在**PIM\_READY**有效时，采样**CNT\_OUT**中对应每一列计数器的最低位，若计数器输出为1则行对应列上的RRAM为高阻态，是0则为低阻态；每次能并行读取一行RRAM的状态。