

目 录

1.	技术规范.....	2
1.1	系统结构	2
1.2	功能要求概述	2
1.3	性能要求	2
1.4	实现载体及相关工具	2
2.	总体设计方案.....	3
2.1	PE ARRAY 电路结构.....	3
2.2	PE ARRAY 电路接口说明.....	3
2.3	时序说明	4
2.4	寄存器说明	4
2.5	PE ARRAY 电路子模块功能说明.....	4
3.	详细设计方案.....	5
3.1	PE 模块设计方案.....	5
3.2	shift_reg 模块设计方案	6
3.3	FSM 模块设计方案	7
4.	验证方案.....	9
4.1	PE_Array 电路仿真平台搭建.....	9
4.2	部分 Makefile	9
4.3	验证用例	9
5.	仿真结果及分析.....	11
5.1	各模块仿真结果	11
5.2	总体仿真结果	12
5.3	覆盖率统计	13
6.	综合结果及分析.....	15
6.1	时序报告	15
6.2	面积报告	15
7.	静态时序分析结果及分析.....	16
8.	形式化验证结果及分析.....	16

1. 技术规范

本设计旨在实现可重构的脉动计算阵列，用于计算矩阵乘法或卷积神经网络。矩阵大小、数据位宽参数均可设置。电路工作频率为 200MHz 以上。

1.1 系统结构

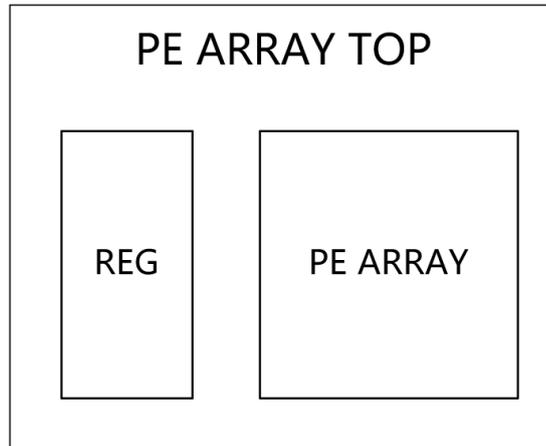


图 1 PE_ARRAY 电路与系统其他单元关系图

1.2 功能要求概述

1. 能够运用脉动阵列原理准确计算矩阵乘法的正确结果。
2. 原始数据和中间计算结果的数据流动正确。
3. 在卷积神经网络的计算中能够连续输入图像数据进行计算

1.3 性能要求

使用 55nm 工艺库，设计电路工作频率为 200Mhz 以上，时序、面积、功耗尽可能平衡优化。

1.4 实现载体及相关工具

本设计采用 smic 55nm 工艺库，采用 Spyglass、VCS、Design Compiler、Formality、Prime Time 等前端工具进行辅助设计、仿真验证。

2. 总体设计方案

2.1 PE ARRAY 电路结构

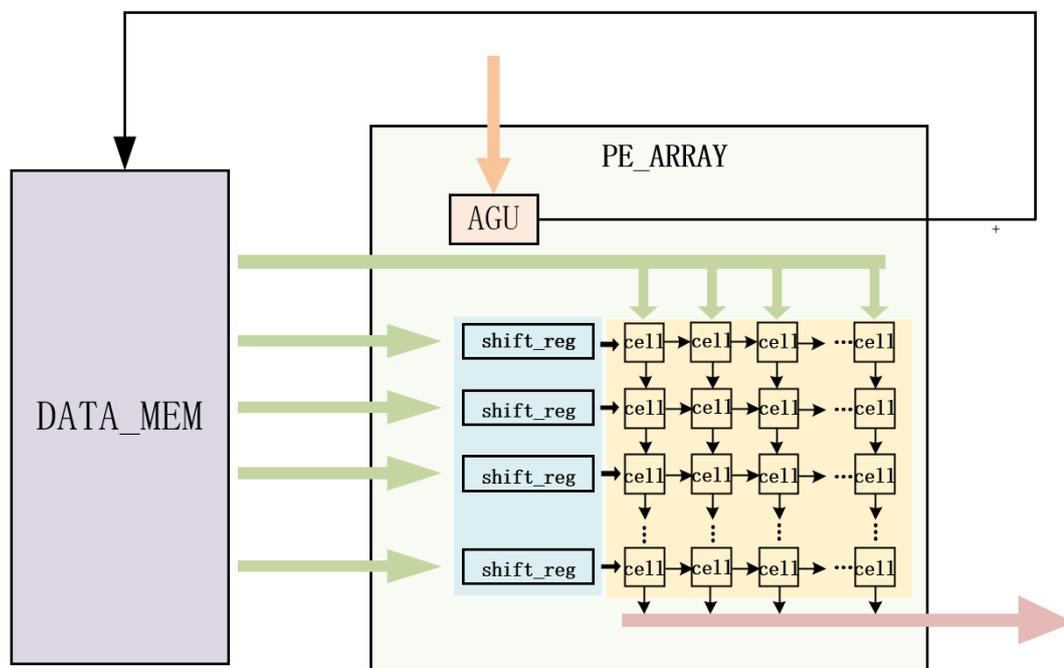


图 2 PE ARRAY 电路结构图

2.2 PE ARRAY 电路接口说明

表 2 PE ARRAY 电路输入输出信号

信号名称	方向	宽度	说明
clk	input	1	时钟信号
rst_n	input	1	复位信号
d_base_addr	input	ADDR_WIDTH	图像数据基地址
w_base_addr	input	ADDR_WIDTH	权重数据基地址
array_in	input	MEM_WIDTH	矩阵数据输入
layer_size	input	MEM_WIDTH	矩阵大小输入
array_result	output	DATA_WIDTH*ARRAYSIZE	矩阵计算结果
addr_out	output	ADDR_WIDTH	数据地址
data_ready	output	1	输出数据有效
done	output	1	计算完成

2.3 时序说明

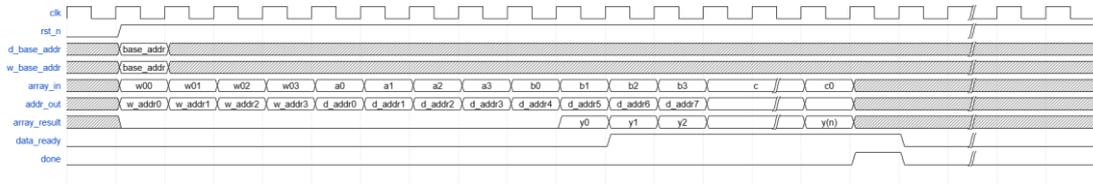


图 3 PE Array 读写时序图

2.4 寄存器说明

表 3 寄存器说明

寄存器名称	宽度
layer_size	MEM_WIDTH

2.4.1 寄存器列表

表 4 pe array 电路寄存器说明

寄存器名	地址偏移	说明
layer_size	无	输入计算矩阵大小
d_base_addr	0xF0000000	保存图像数据块基地址
w_base_addr	0xF0000001	保存权重数据块基地址

2.5 PE ARRAY 电路子模块功能说明

本设计要求

子模块 pe 实现：对输入的权重数据进行寄存直到下一个权重寄存，对输入的数据与寄存的权重进行乘操作，实现乘法结果加一个加法结果输入，并将这个乘加结果输出。

子模块 shift register 实现：对输入的 32bit 数据进行寄存，并按 8bit 步长移位输出低位。

子模块状态机实现：将输入权重数据、输入图像数据、流动计算、计算完成的整个过程做成状态机，输出每个状态的控制信号。

3. 详细设计方案

3.1 PE 模块设计方案

3.1.1 PE 模块功能描述

矩阵乘法器的单个计算核心，接受两个操作数与上层的计算结果进行计算，将正群结果继续向下层流动

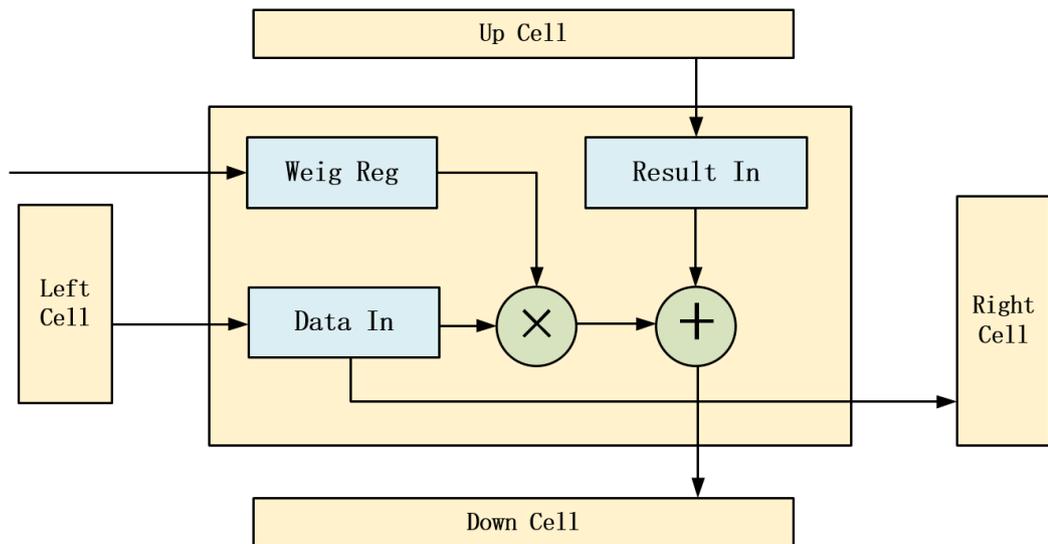


图 4 PE 模块结构图

3.1.2 PE 模块接口说明

表 5 PE 子模块 1 输入输出信号说明

信号名称	方向	宽度	说明
clk	input	1	时钟信号
rst_n	input	1	复位信号
data_in	input	DATA_WIDTH	PE 图像数据输入
weith_in	input	DATA_WIDTH	PE 权重数据输入
result_in	input	DATA_WIDTH	上层 PE 结果
data_flow	output	DATA_WIDTH	图像数据横向流动输出
reslut_flow	output	DATA_WIDTH	计算结果纵向流动输出

3.1.3 PE 模块时序说明

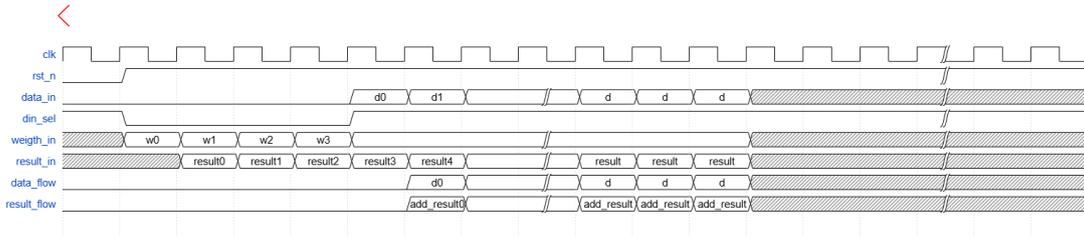


图 5 PE 模块时序图

3.2 shift_reg 模块设计方案

3.2.1 shift_reg 模块功能描述

旨在解决存储器存储数据位宽和 PE 阵列数据输入位宽不匹配问题。通过添加移位寄存器，将存储器中同一个地址中存放的数据的不同字段拆分，按序输入到 PE 阵列的数据输入端口中。

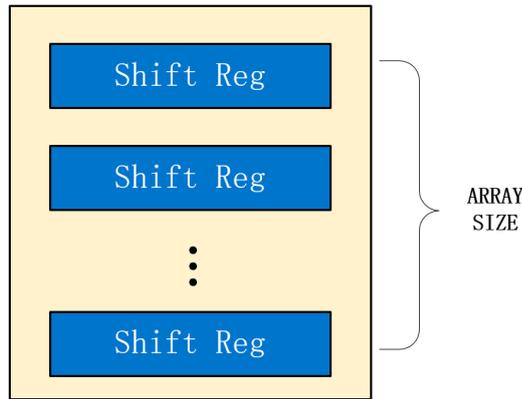


图 4 shift_reg 模块结构图

3.2.2 shift_reg 模块接口说明

表 5 子模块 1 输入输出信号说明

信号名称	方向	宽度	说明
clk	input	1	时钟信号
rst_n	input	1	复位信号
wr	input	1	移位寄存器写使能
din	input	MEM_WIDTH	从存储器中输入的数据
dout	input	DATA_WIDTH	移位寄存器输出

3.2.3 shift_reg 模块时序图

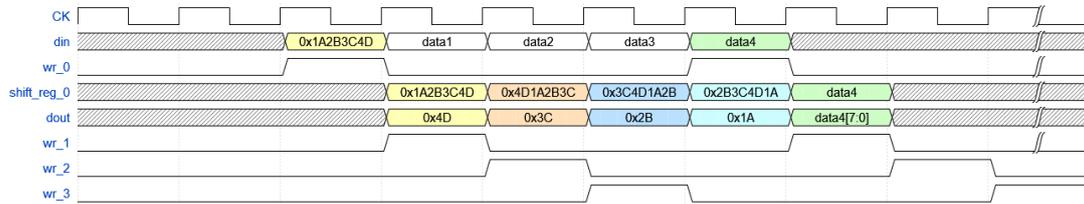


图 6 shift_reg 模块时序图

3.3 FSM 模块设计方案

3.3.1 FSM 模块功能描述

将输入权重数据、输入图像数据、流动计算、计算完成的整个过程做成状态机，输出每个状态的控制信号。

3.3.2 FSM 模块接口说明

表 5 FSM 输入输出信号说明

信号名称	方向	宽度	说明
clk	input	1	时钟信号
rst_n	input	1	复位信号
data_ready	output	1	标志开始计算出结果
din_sel	output	1	标志数据为图像/权重
done	output	1	标志整体计算完成

3.3.3 FSM 模块状态转移图

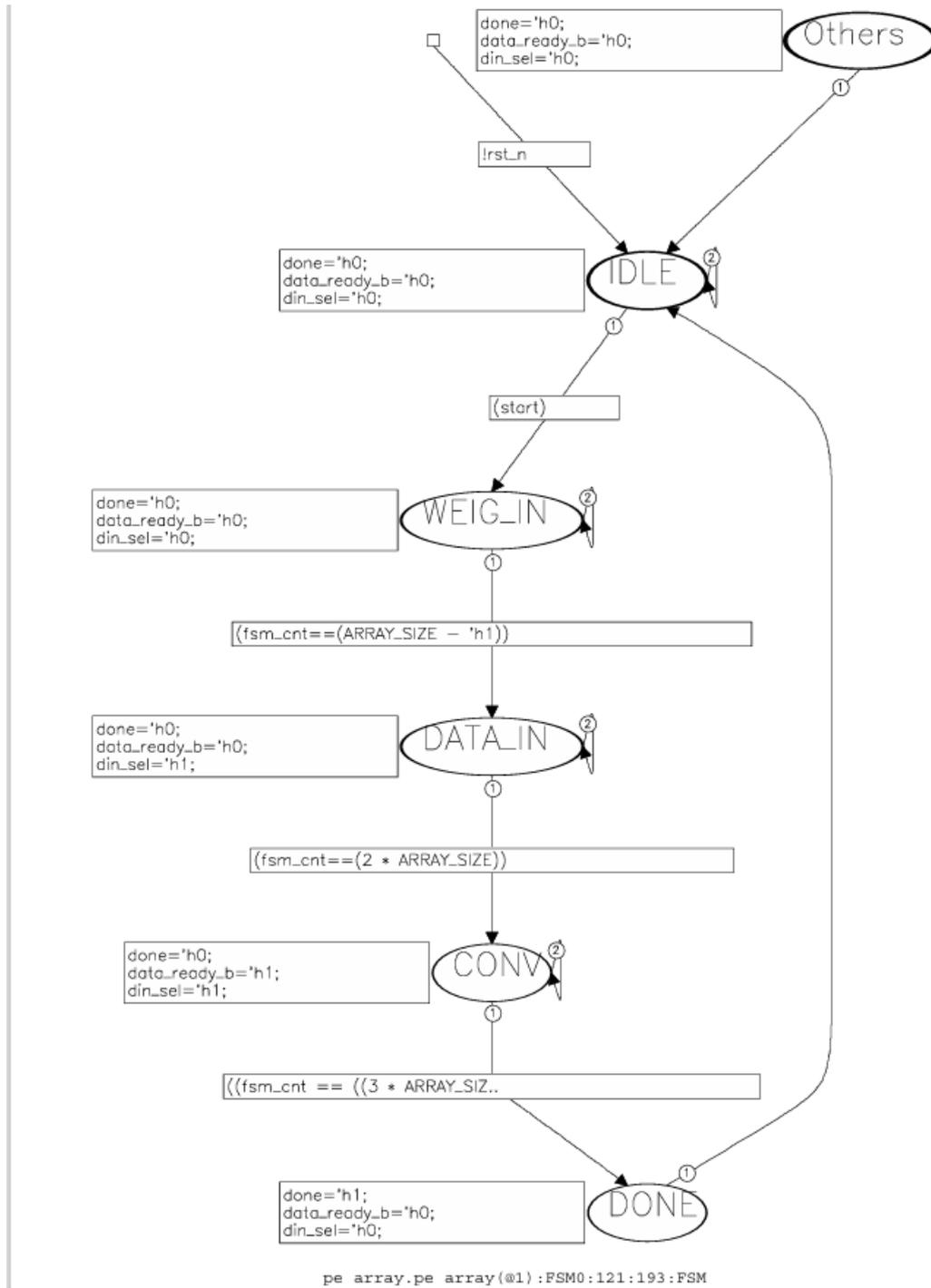


图 7 veridi 生成的状态转移图

4. 验证方案

4.1 PE_Array 电路仿真平台搭建

编写测试激励，使用 VCS 进行编译仿真。

为了仿真流程的自动化程度，使用 Makefile 脚本进行快捷编译、仿真、加载波形，极大提高了仿真的效率。

同时，为了仿真测试激励的全面性，封装 task 任务，可以重复调用，提高仿真测试激励编写的效率，使仿真激励尽可能的高效。

4.2 部分 Makefile

```
all :rvcs rsim rdve
```

```
rvcs :
```

```
vcs -full64 -debug_access+all -sverilog -f ../script/filelist.f +incdir+../tb+ ../tb/tb.sv -  
timescale=1ns/1ps +memcbk -l vcs.log
```

```
rsim :
```

```
./simv -l sim.log
```

```
rdve :
```

```
dve -vpd *.vpd -script *.tcl
```

4.3 验证用例

```
// pe_array Parameters
```

```
parameter PERIOD      = 10;
```

```
parameter ADDR_WIDTH  = 32;
```

```
parameter ARRAY_SIZE  = 4 ;
```

```
parameter DATA_WIDTH = 8 ;
```

```
parameter MEM_WIDTH   = 32;
```

```
class array_rand;
```

```
    rand  bit [31:0] weigth_rand_in;
```

```
    rand  bit [31:0] data_rand_in;
```

```
endclass
```

```
array_rand array_in_rand  = new();
```

```

task initialize_array(output [MEM_WIDTH-1:0] initialize_out);

    if(i<ARRAY_SIZE) begin
        array_in_rand.randomize();
        initialize_out = array_in_rand.weigth_rand_in;
        $display("Weigth Initialization %h", initialize_out);
    end
    if(i == ARRAY_SIZE) begin
        $display("*****");
        initialize_out = 'h0;
    end
    if(i>=ARRAY_SIZE+1) begin
        array_in_rand.randomize();
        initialize_out = array_in_rand.data_rand_in;
        $display("Data Initialization %h", initialize_out);
    end
    i++;

endtask

initial begin
    wait(rst_n);

    $display("*****");
    $display("*****Initialization*****\n");

    repeat (2*ARRAY_SIZE+1)@(posedge clk) initialize_array(array_in_pe);

    $display("*****Initialization finished*****");
    $display("*****");
end

```

使用随机约束仿真，利用 task 封装写入数据过程，写权重与写数据分先后写入。在例化调用 task 时，可以使用 repeat 来重复调用任意次，大大提高了仿真效率。利用 System Verilog 部分特性可以使仿真更加的直观与便捷。

5. 仿真结果及分析

5.1 各模块仿真结果

5.1.1 移位寄存器仿真

移位寄存器的功能是为 pe_array 中的每一横排提供图像数据。每一个移位寄存器中寄存的数据为一横排 pe 所需图像数据，从最左一列 pe 进入，在 pe_array 内部不断向右移位。因此移位寄存器从输入图像数据的状态开始，使能应该循环拉高，并且内部数据不断移位，在每次使能数据有效后更新数据。

图为 4*4 大小 pe_array 移位寄存器仿真结果，可以看到移位寄存器的使能是正确循环拉高的，并且内部移位正确，状态机正常跳转，说明移位寄存器的工作节点正确，是从进入数据开始工作的。

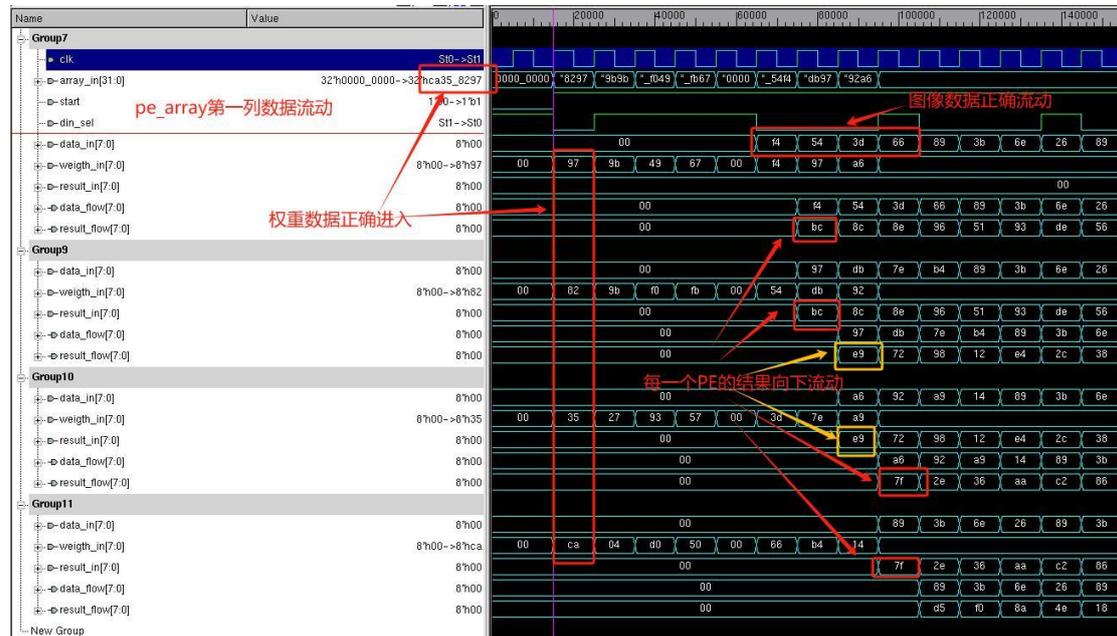


5.1.2 pe_array 内部数据流动仿真

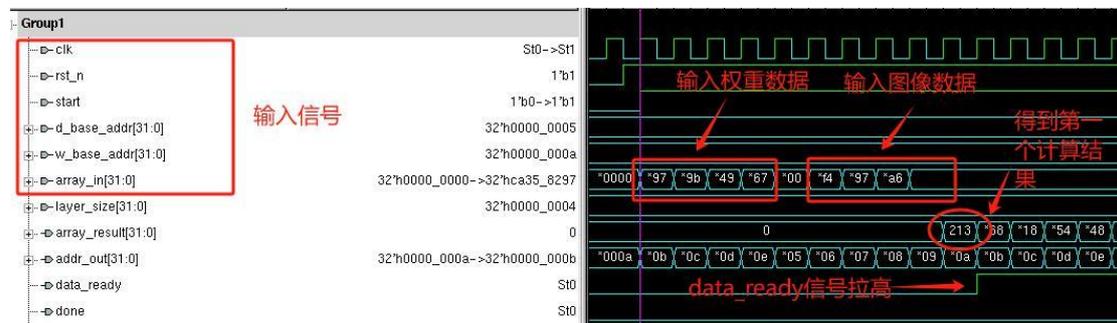
在 pe_array 中，图像数据不断向右流动，即每一次参与计算的图像数据在计算完成后向右传递给下一个 cell；计算结果不断向下流动，即每一个 cell 计算出的结果向下传递给下一个 cell，最后在整个阵列的最下面一行输出结果。数据的纵横流动构成了整个脉动阵列。

图为 4*4 大小 pe_array 内部数据流动仿真结果。本次仿真验证了第一列 pe 的数据流动，移位寄存器的输出接入最左一列 pe 单元，输入数据正确，同时权重数据也正确进入。经过 pe 单元计算后，计算结果正确，结果正确流入下一个

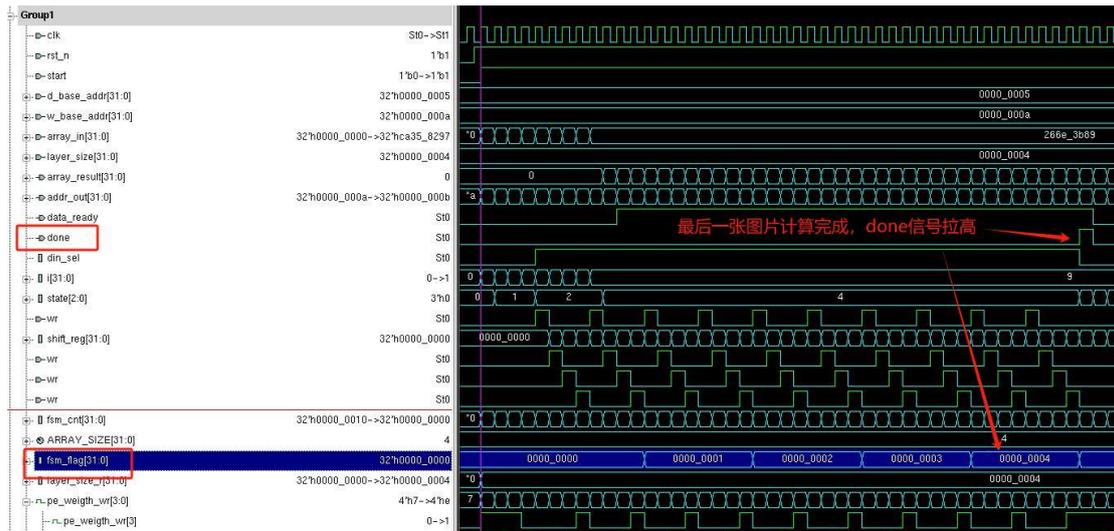
cell 中，数据流动符合预期结果。



5.2 总体仿真结果



图为总体仿真结果，可以看到在正确给出 start 信号、图像数据和权重数据的基地址和数据后，权重数据和图像数据正确进入 pe_array，并在正确得出结果，data_ready（单个 cell 结果计算完成）信号正确拉高。



经过连续计算，fsm_flag（图像数 flag，即连续计算 flag 次）信号正确计算，在最后一张图片计算完成后，输出一个 done 信号。整体数据流动和计算结果正确。

5.3 覆盖率统计

Name	Score	Line	FSM	Condition
array_tb	90.57%	98.99%	72.73%	100.00%
u_pe_array	90.52%	98.82%	72.73%	100.00%
pe_x[0].pe_y[0].u_pe	100.00%	100.00%		
pe_x[0].pe_y[1].u_pe	100.00%	100.00%		
pe_x[0].pe_y[2].u_pe	100.00%	100.00%		
pe_x[0].pe_y[3].u_pe	100.00%	100.00%		
pe_x[1].pe_y[0].u_pe	100.00%	100.00%		
pe_x[1].pe_y[1].u_pe	100.00%	100.00%		
pe_x[1].pe_y[2].u_pe	100.00%	100.00%		
pe_x[1].pe_y[3].u_pe	100.00%	100.00%		
pe_x[2].pe_y[0].u_pe	100.00%	100.00%		
pe_x[2].pe_y[1].u_pe	100.00%	100.00%		
pe_x[2].pe_y[2].u_pe	100.00%	100.00%		
pe_x[2].pe_y[3].u_pe	100.00%	100.00%		
pe_x[3].pe_y[0].u_pe	100.00%	100.00%		
pe_x[3].pe_y[1].u_pe	100.00%	100.00%		
pe_x[3].pe_y[2].u_pe	100.00%	100.00%		
pe_x[3].pe_y[3].u_pe	100.00%	100.00%		
shift_reg_in[0].u_shift_r...	100.00%	100.00%		
shift_reg_in[1].u_shift_r...	100.00%	100.00%		
shift_reg_in[2].u_shift_r...	100.00%	100.00%		
shift_reg_in[3].u_shift_r...	100.00%	100.00%		

整体覆盖率统计如图，行覆盖率达到 98.99%，状态机覆盖率为 72.73%，条件覆盖率达到 100%。

```

3
0   DATA_IN:   begin
1   done = 'h0;
2   data_ready_b = 'h0;
3   din_sel = 'h1;
4   end
5
6   CONV:       begin
7   done = 'h0;
8   data_ready_b = 'h1;
9   din_sel = 'h1;
0   end
1
2   DONE:       begin
3   done = 'h1;
4   data_ready_b = 'h0;
5   din_sel = 'h0;
6   end
7   default :   begin
8   done = 'h0;
9   data_ready_b = 'h0;
0   din_sel = 'h0;
1   end
2   endcase

```

状态机共有 5 个状态，需要 3bit 信号定义 state，因此 state 信号一定有非法值。在代码中给出了 default 值，使状态机错误时可以复位。在仿真中这一情况未覆盖到。

<input type="checkbox"/> Shown in List					
	✓ 0	✓ 1	✓ 2	✓ 3	✓ 4
✓ 0 IDLE	-	✓	-	-	-
✓ 1 WEIG_IN	✗	-	✓	-	-
✓ 2 DATA_IN	✗	-	-	✓	-
✓ 3 CONV	✗	-	-	-	✓
✓ 4 DONE	✓	-	-	-	-

在状态的跳转中，并不会出现从 WEIG_IN/DATA_IN/CONV 这几个状态中跳转回 IDLE，因此状态机覆盖率覆盖不到。

其他所有的代码、条件、状态均被覆盖。

6. 综合结果及分析

6.1 时序报告

```

*****
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : pe_array
Version: 0-2018.06-SP1
Date : Tue Jun 11 09:02:22 2024
*****
Operating Conditions: tt_vlp2_25c Library: scc55nll_hd_r
Wire Load Model Mode: top

Startpoint: fsm_cnt_reg_0
(rising edge-triggered flip-flop clocked by
Endpoint: fsm_cnt_reg_31
(rising edge-triggered flip-flop clocked by cl
Path Group: clk
Path Type: max

Point                               Incr      Path
-----
clock clk (rise edge)                0.00      0.00
clock network delay (ideal)          0.00      0.00
fsm_cnt_reg_0 /CK (DRNQH0V1)         0.00      0.00
fsm_cnt_reg_0 /Q (DRNQH0V1)         0.17      0.17
U528/Z/N (NAND2HDV0)                 0.05      0.22
U505/Z/N (NOR2HDV1)                  0.05      0.27
U479/Z/N (CLKNAND2HDV1)              0.05      0.33
U458/Z/N (NOR2HDV1)                  0.07      0.40
U453/Z/N (CLKNAND2HDV1)              0.06      0.46
U439/Z/N (NOR2HDV1)                  0.07      0.53
U434/Z/N (CLKNAND2HDV1)              0.06      0.59
U428/Z/N (NOR2HDV1)                  0.07      0.66
U424/Z/N (CLKNAND2HDV1)              0.06      0.72
U419/Z/N (NOR2HDV1)                  0.07      0.79
U413/Z/N (CLKNAND2HDV1)              0.06      0.85
U405/Z/N (NOR2HDV1)                  0.07      0.92
U403/Z/N (CLKNAND2HDV1)              0.06      0.98
U395/Z/N (NOR2HDV1)                  0.07      1.05
U389/Z/N (CLKNAND2HDV1)              0.06      1.11
U377/Z/N (NOR2HDV1)                  0.07      1.18
U364/Z/N (CLKNAND2HDV1)              0.06      1.24
U357/Z/N (NOR2HDV1)                  0.07      1.31
U351/Z/N (CLKNAND2HDV1)              0.06      1.37
U346/Z/N (NOR2HDV1)                  0.07      1.44
U338/Z/N (CLKNAND2HDV1)              0.06      1.50
U334/Z/N (NOR2HDV1)                  0.07      1.57
U330/Z/N (CLKNAND2HDV1)              0.06      1.63
U320/Z/N (NOR2HDV1)                  0.07      1.70
U315/Z/N (CLKNAND2HDV1)              0.06      1.76
U312/Z/N (NOR2HDV1)                  0.07      1.83
U304/Z/N (CLKNAND2HDV1)              0.05      1.88
U746/Z (CLKXOR2HDV2)                 0.07      1.96
fsm_cnt_reg_31 /D (DRNQH0V1)        0.00      1.96
data arrival time                     0.00      1.96
data required time                    2.50      2.50
clock network delay (ideal)          0.00      2.50
clock uncertainty                      -0.30     2.20
fsm_cnt_reg_31 /CK (DRNQH0V1)        0.00      2.20
library setup time                    -0.07     2.13
data required time                    2.13
data arrival time                     -1.96
slack (MET)                           0.17

```

添加时序约束后,可以通过 DC 时序报告看出,综合阶段频率可以达到 400MHz,最长路径建立时间符合约束。

6.2 面积报告

Hierarchical area distribution						
Hierarchical cell	Global cell area		Local cell area			Design
	Absolute Total	Percent Total	Combi-national	Noncombi-national	Black-boxes	
pe_array	10624.3198	100.0	1478.6800	1129.5200	0.0000	pe_array
pe_x_0_pe_y_0_u_pe	404.0400	3.8	242.7600	161.2800	0.0000	pe_DATA_WIDTH8_15
pe_x_0_pe_y_1_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_14
pe_x_0_pe_y_2_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_13
pe_x_0_pe_y_3_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_12
pe_x_1_pe_y_0_u_pe	404.0400	3.8	242.7600	161.2800	0.0000	pe_DATA_WIDTH8_11
pe_x_1_pe_y_1_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_10
pe_x_1_pe_y_2_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_9
pe_x_1_pe_y_3_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_8
pe_x_2_pe_y_0_u_pe	404.0400	3.8	242.7600	161.2800	0.0000	pe_DATA_WIDTH8_7
pe_x_2_pe_y_1_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_6
pe_x_2_pe_y_2_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_5
pe_x_2_pe_y_3_u_pe	453.3200	4.3	292.0400	161.2800	0.0000	pe_DATA_WIDTH8_4
pe_x_3_pe_y_0_u_pe	341.3200	3.2	233.8000	107.5200	0.0000	pe_DATA_WIDTH8_3
pe_x_3_pe_y_1_u_pe	397.0400	3.7	289.5200	107.5200	0.0000	pe_DATA_WIDTH8_2
pe_x_3_pe_y_2_u_pe	395.9200	3.7	288.4000	107.5200	0.0000	pe_DATA_WIDTH8_1
pe_x_3_pe_y_3_u_pe	395.9200	3.7	288.4000	107.5200	0.0000	pe_DATA_WIDTH8_0
shift_reg_in_0_u_shift_reg_din	298.4800	2.8	83.4400	215.0400	0.0000	shift_reg_din_DATA_WIDTH8_MEM_WIDTH32_3
shift_reg_in_1_u_shift_reg_din	298.4800	2.8	83.4400	215.0400	0.0000	shift_reg_din_DATA_WIDTH8_MEM_WIDTH32_2
shift_reg_in_2_u_shift_reg_din	298.4800	2.8	83.4400	215.0400	0.0000	shift_reg_din_DATA_WIDTH8_MEM_WIDTH32_1
shift_reg_in_3_u_shift_reg_din	298.4800	2.8	83.4400	215.0400	0.0000	shift_reg_din_DATA_WIDTH8_MEM_WIDTH32_0
Total			6269.2000	4355.1199	0.0000	

通过面积报告可以看出，综合阶段使用 55nm 工艺库面积为 10624.3198um²，可以看出共有 16 个 pe cell，4 个 shift_reg cell。

7. 静态时序分析结果及分析

```

*****
Report : timing
  -path_type full
  -delay_type max
  -max_paths 1
  -sort_by slack
Design : pe_array
Version: 0-2018.06-SP1
Date : Tue Jun 11 09:44:54 2024
*****

Startpoint: fsm_flag_reg_0
(rising edge-triggered flip-flop clocked by clk)
Endpoint: fsm_flag_reg_31_
(rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Point                               Incr      Path
-----
clock clk (rise edge)                0.00      0.00
clock network delay (ideal)          0.00      0.00
fsm_flag_reg_0 /CK (DRNQH0V1)        0.00      0.00 r
fsm_flag_reg_0 /Q (DRNQH0V1)        0.16 *    0.16 r
U549/ZN (CLKNAND2HDV1)               0.05 *    0.21 f
U507/ZN (NOR2HDV1)                   0.06 *    0.27 r
U464/ZN (CLKNAND2HDV1)               0.05 *    0.32 f
U461/ZN (NOR2HDV1)                   0.07 *    0.39 r
U450/ZN (CLKNAND2HDV1)               0.06 *    0.45 f
U443/ZN (NOR2HDV1)                   0.07 *    0.52 r
U436/ZN (CLKNAND2HDV1)               0.06 *    0.58 f
U429/ZN (NOR2HDV1)                   0.07 *    0.65 r
U423/ZN (CLKNAND2HDV1)               0.06 *    0.71 f
U417/ZN (NOR2HDV1)                   0.07 *    0.78 r
U415/ZN (CLKNAND2HDV1)               0.06 *    0.84 f
U408/ZN (NOR2HDV1)                   0.07 *    0.91 r
U401/ZN (CLKNAND2HDV1)               0.06 *    0.97 f
U398/ZN (NOR2HDV1)                   0.07 *    1.05 f
U394/ZN (CLKNAND2HDV1)               0.06 *    1.10 f
U383/ZN (NOR2HDV1)                   0.07 *    1.18 r
U362/ZN (CLKNAND2HDV1)               0.06 *    1.24 f
U356/ZN (NOR2HDV1)                   0.07 *    1.31 r
U349/ZN (CLKNAND2HDV1)               0.06 *    1.37 f
U344/ZN (NOR2HDV1)                   0.07 *    1.44 r
U342/ZN (CLKNAND2HDV1)               0.06 *    1.50 f
U332/ZN (NOR2HDV1)                   0.07 *    1.57 r
U325/ZN (CLKNAND2HDV1)               0.06 *    1.63 f
U323/ZN (NOR2HDV1)                   0.07 *    1.70 r
U318/ZN (CLKNAND2HDV1)               0.06 *    1.76 f
U311/ZN (NOR2HDV1)                   0.07 *    1.83 r
U302/ZN (CLKNAND2HDV1)               0.05 *    1.88 f
U730/Z (CLKKOR2HDV2)                 0.08 *    1.96 r
U297/Z (AO22HDV1)                   0.06 *    2.02 r
fsm_flag_reg_31 /D (DRNQH0V1)       0.00 *    2.02 r
data arrival time                    0.00 *    2.02
-----
clock clk (rise edge)                2.50      2.50
clock network delay (ideal)          0.00      2.50
clock reconvergence pessimism        0.00      2.50
clock uncertainty                     -0.30     2.20
fsm_flag_reg_31 /CK (DRNQH0V1)      0.00 *    2.20 r
library setup time                   -0.07 *   2.13
data required time                    0.00 *    2.13
-----
data required time                    0.00 *    2.13
data arrival time                    -0.00 *   -2.02
-----
slack (MET)                          0.06 *    0.11

```

在进行 PT 静态时序分析时，使用 sdc 文件作为约束进行静态时序分析，在静态时序分析报告可以看出，在进行约束之后的时序分析报告中，最长路径建立时间符合时序。

8. 形式化验证结果及分析

```

***** Verification Results *****
Verification SUCCEEDED

Reference design: r:/WORK/pe_array
Implementation design: i:/WORK/pe_array
714 Passing compare points
-----
Matched Compare Points   BBPin   Loop   BBNet   Cut   Port   DFF   LAT   TOTAL
-----
Passing (equivalent)    0       0     0       0     66    648   0     714
Failing (not equivalent) 0       0     0       0     0     0     0     0
-----
fm_shell (verify)> report_unmatched_points
*****
Report : unmatched_points

Reference : r:/WORK/pe_array
Implementation : i:/WORK/pe_array
Version : K-2015.06-SP1
Date : Tue Jun 11 14:12:41 2024
*****
No unmatched points.

```

将 RTL 文件、Netlist 文件和 Lib 文件 load 到 formality 中，verification 结果为 succeed，所有比较点均匹配，无不匹配点。