

**SOC设计方法大作业报告**

——基于Sobel算子的图像增强

**组长：王鹏飞（3119105033）**

**组员：王佳兴（3119105038）**

**曹卓勋（3119305349）**

**石益鑫（3119305383）**

**目录**

[一、 项目概述 1](#_Toc29931068)

[1. 简介 1](#_Toc29931069)

[2. 实现过程 2](#_Toc29931070)

[2.1功能说明 2](#_Toc29931071)

[2.2引脚说明 2](#_Toc29931072)

[2.3性能指标 3](#_Toc29931073)

[二、 实现步骤 4](#_Toc29931074)

[1. 硬件实现与仿真 4](#_Toc29931075)

[1.1 Matlab建模仿真 4](#_Toc29931076)

[1.2 Verilog代码实现 4](#_Toc29931077)

[2. 综合及网表仿真 6](#_Toc29931078)

[2.1 使用DC综合 6](#_Toc29931079)

[2.2 综合结果 6](#_Toc29931080)

[2.3 综合后仿真 7](#_Toc29931081)

[3. 基于AHB总线的IP核开发及仿真 9](#_Toc29931082)

[4. FPGA验证与仿真 9](#_Toc29931083)

# 项目概述

## 1. 简介

RGB色彩模式是工业界的一种颜色标准，通过对红(R)、绿(G)、蓝(B)三个颜色通道的变化以及叠加来得到各种颜色，是运用最广的颜色系统之一。

YUV颜色编码方法，主要用于优化彩色视频信号的传输。其中Y表示明亮度，也就是灰阶值；而U和V表示色度，作用是描述影像色彩及饱和度，用于指定像素的颜色。

将图片从RGB空间转换到YUV空间，按以下公式实现转换：

Y = 0.299R + 0.587G + 0.114B；

U = -0.169R - 0.331G + 0.5B ;

V = 0.5R - 0.419G - 0.081B;

将图片从YUV空间转换到RGB空间，按以下公式实现转换：

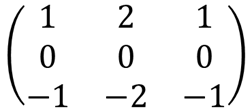
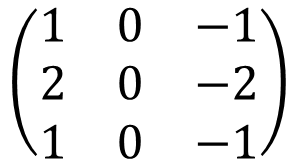
R = Y + 1.4075 \* (V-128);

G = Y - 0.3455 \* (U-128) - 0.7169\*(V-128);

B = Y + 1.779 \* (U-128);

图像边缘指的是图像中灰度值或亮度发生明显变化的局部区域。边缘检测是图像处理和计算机视觉中，尤其是特征提取中的一个重要研究领域。Sobel 算子是一种实现边缘检测功能的简单算子。

坐标为的像素点与其周围像素点的灰度表示为以下矩阵：

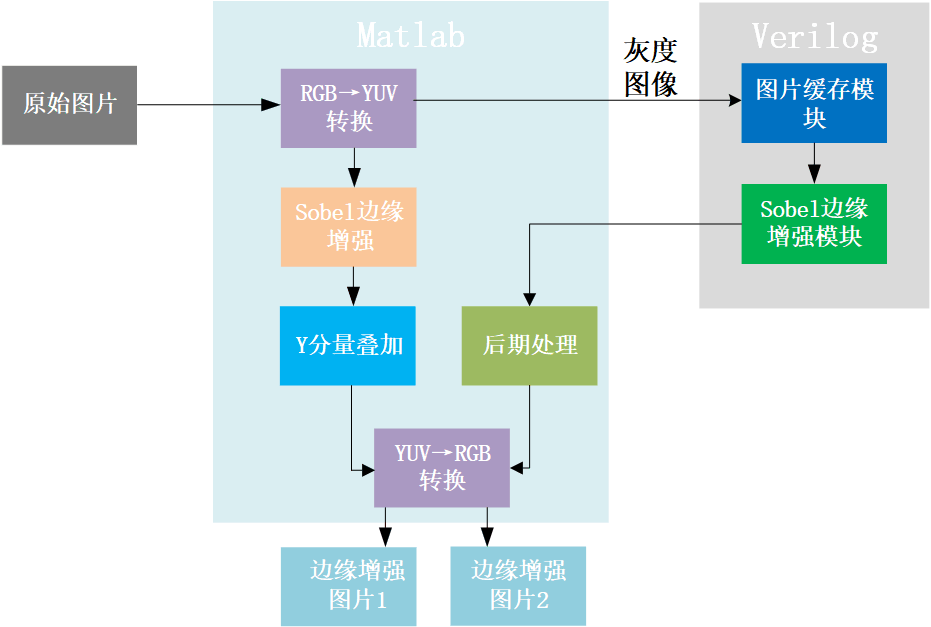
Sobel算子是离散微分算子，利用3\*3 卷积模板，通过计算目标像素点水平和垂直方向上的亮度差分计算图像灰度的近似梯度，从而实现边缘检测功能，水平方向和垂直方向算子矩阵分别是：

灰度值沿水平和垂直方向上的变化率由卷积运算求解得到：

梯度

## 2. 实现过程

### 2.1功能说明

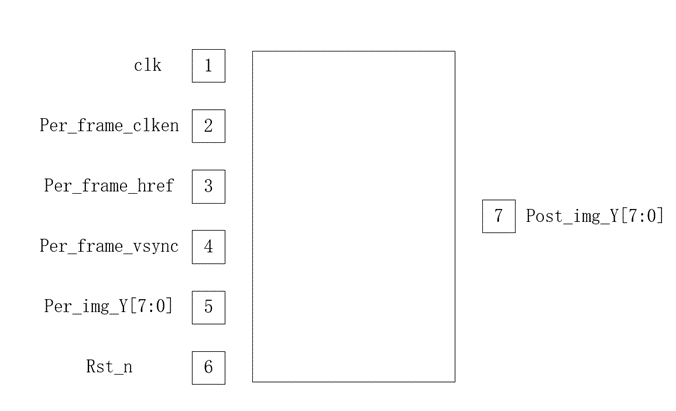
本项目利用Matlab将图片从RGB空间转换为YUV空间，然后利用本项目设计的Sobel算子边缘增强算法模块对灰度Y上进行边缘增强，结果叠加在原Y值上。然后再利用Matlab将YUV图像转换为RGB图像。

支持输入8位RGB值；

输出为8位灰度格式；

* 1. V的数字电源、3.3 V I/O电源

### 2.2引脚说明

本项目设计的IP核的引脚下图所示：

|  |  |  |  |
| --- | --- | --- | --- |
| Pin No | Mnemonic | Type | Function |
| 1 | Clk | I | 时钟信号 |
| 2 | Per\_frame\_clken | I | 数据使能信号，高电平有效 |
| 3 | Per\_frame\_href | I | 行同步信号 |
| 4 | Per\_frame\_vsync | I | 场同步信号 |
| 5 | Per\_img\_Y[7:0] | I | 输入灰度图像，位宽8位 |
| 6 | Rst\_n | I | 复位信号 |
| 7 | Post\_img\_Y[7:0] | O | 输出灰度图像，位宽8位 |

### 2.3性能指标

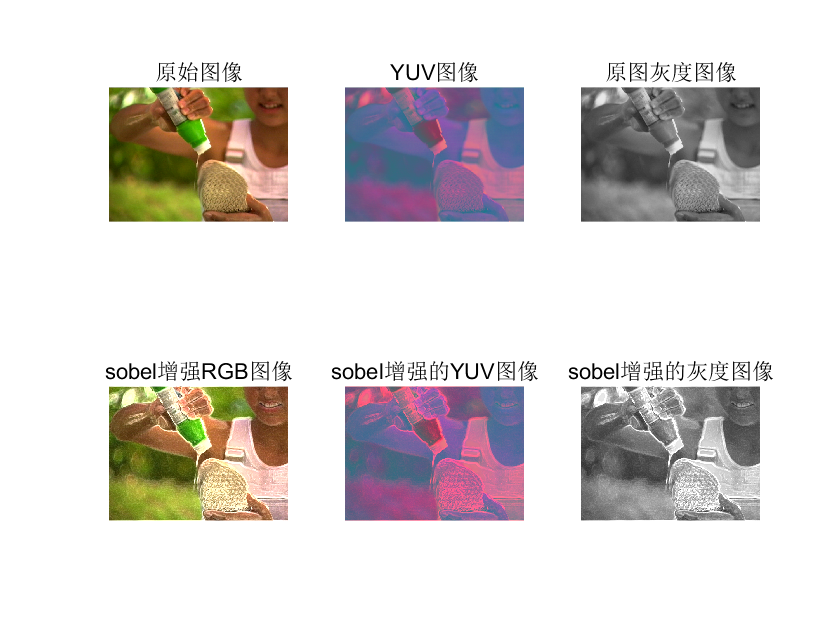
VDD = 3.3 V external/internal VDRIVE = 1.8V, 时钟100Mhz

|  |  |  |
| --- | --- | --- |
| parameter |  |  |
| data arrival time |  |  |
| data required time |  |  |
| slack(MET) |  |  |
|  |  |  |
| cell area |  |  |
| total area |  |  |
|  |  |  |
| input voltage |  |  |
| output voltage |  |  |
|  |  |  |
| input high voltage |  |  |
| input low voltage |  |  |
|  |  |  |
| Output High Voltage (VOH) |  |  |
| Output Low Voltage (VOL) |  |  |

# 实现步骤

## 1. 硬件实现与仿真

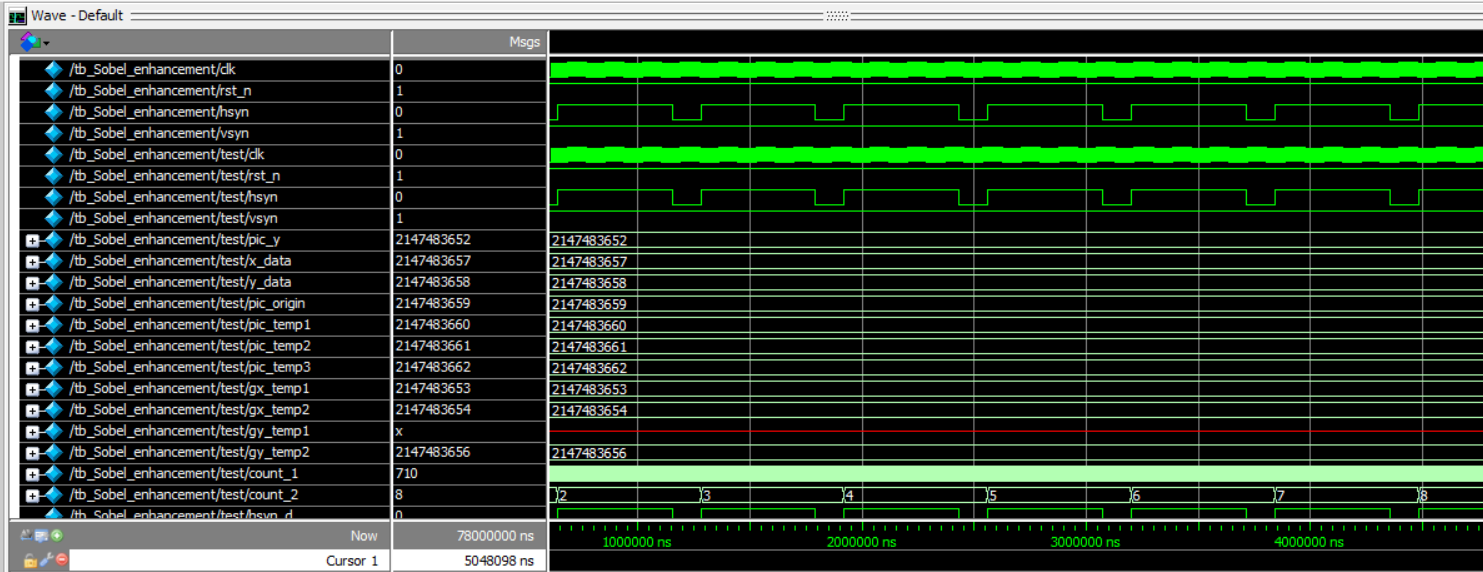
### 1.1 Matlab建模仿真

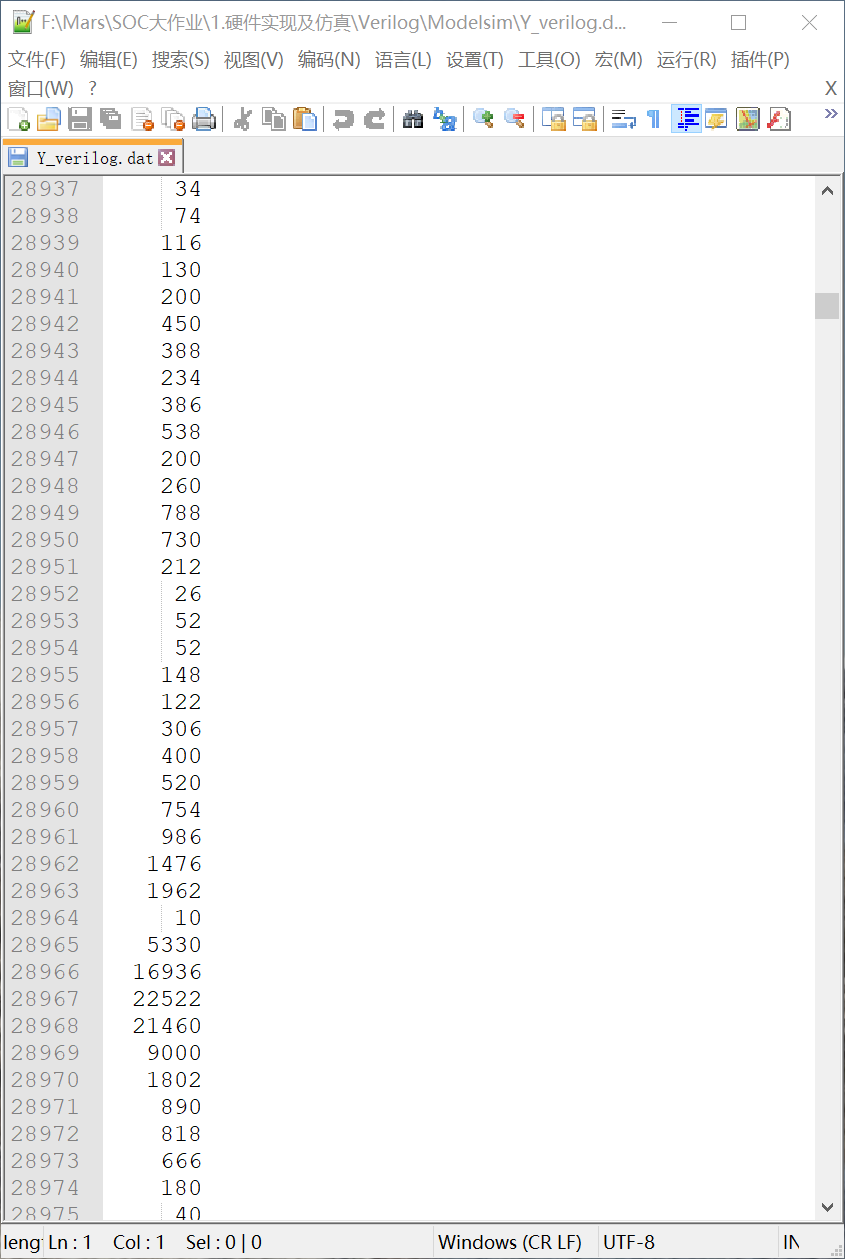
使用Matlab进行功能仿真，对Sobel算子边缘增强进行验证。编写Matlab程序将提供的测试图片读入，RGB转为YUV格式，对灰度进行Sobel增强，叠加入原灰度中，然后YUV格式转为RGB，显示生成图片。由Matlab仿真产生的结果如下图：

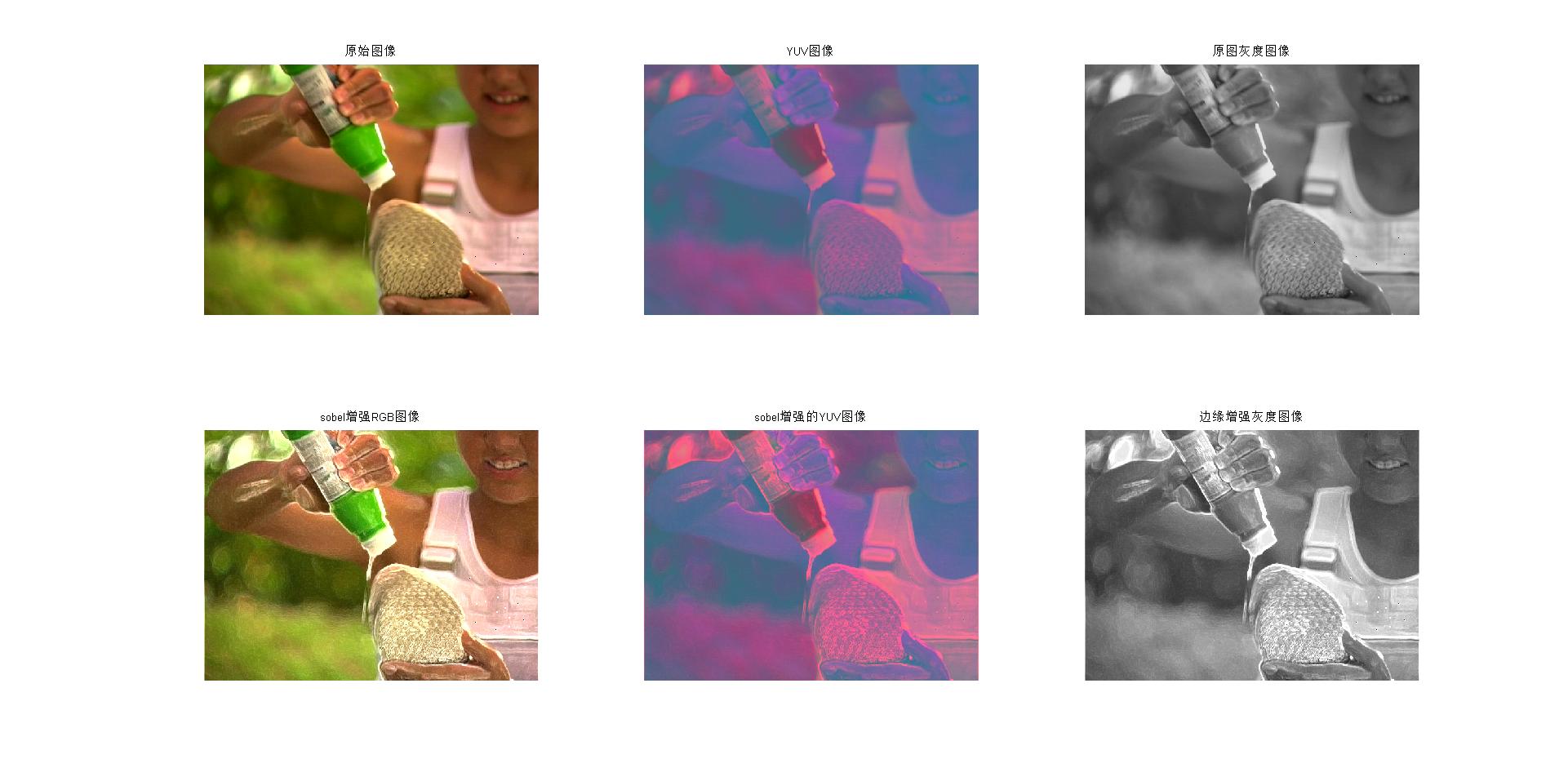
### 1.2 Verilog代码实现

Verilog代码的思路是，首先将整张图片读入。然后再对整张图像逐个像素点进行处理，边处理边输出。由于Modelsim不能进行开方运算，为了保证计算的精度，所以直接将平方和作为结果输出，利用Matlab再进行后期处理。

编写testbench，使用Modelsim仿真工具，得到波形如下所示:



输出的灰度数据如下图所示：

利用Matlab处理得到的RGB图像如下图所示：

可以看出Verilog编写的程序用Modelsim仿真的结果与Matlab边缘增强程序生成的结果相符合。由于NCverilog的simulator工具没有破解成功，因此没有NCverilog仿真结果。

## 2. 综合及网表仿真

### 2.1 使用DC综合

本设计采用synopsys 公司的 ASIC 综合器产品 Design Compiler（简称 DC），它可以完成将硬件描述语言所做的 RTL 级描述自动转换成优化的门级网表。使用DC工具进行综合的流程如下;

（1）检查启动文件，设置库文件；

set link\_library smic18\_tt.db

set target\_library smic18\_tt.db

set symbol\_library smic18.sdb

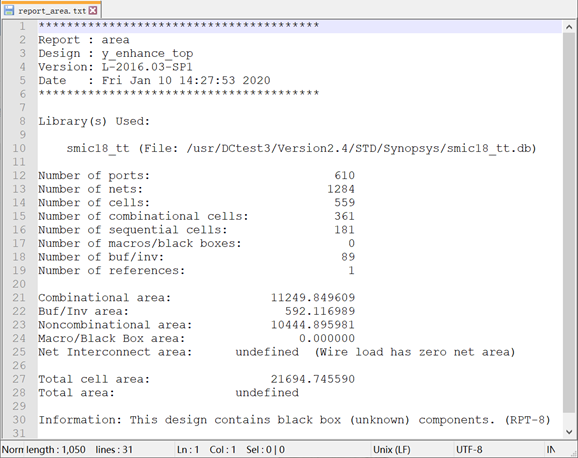
（2）启动DC，读入可综合的Verilog代码；

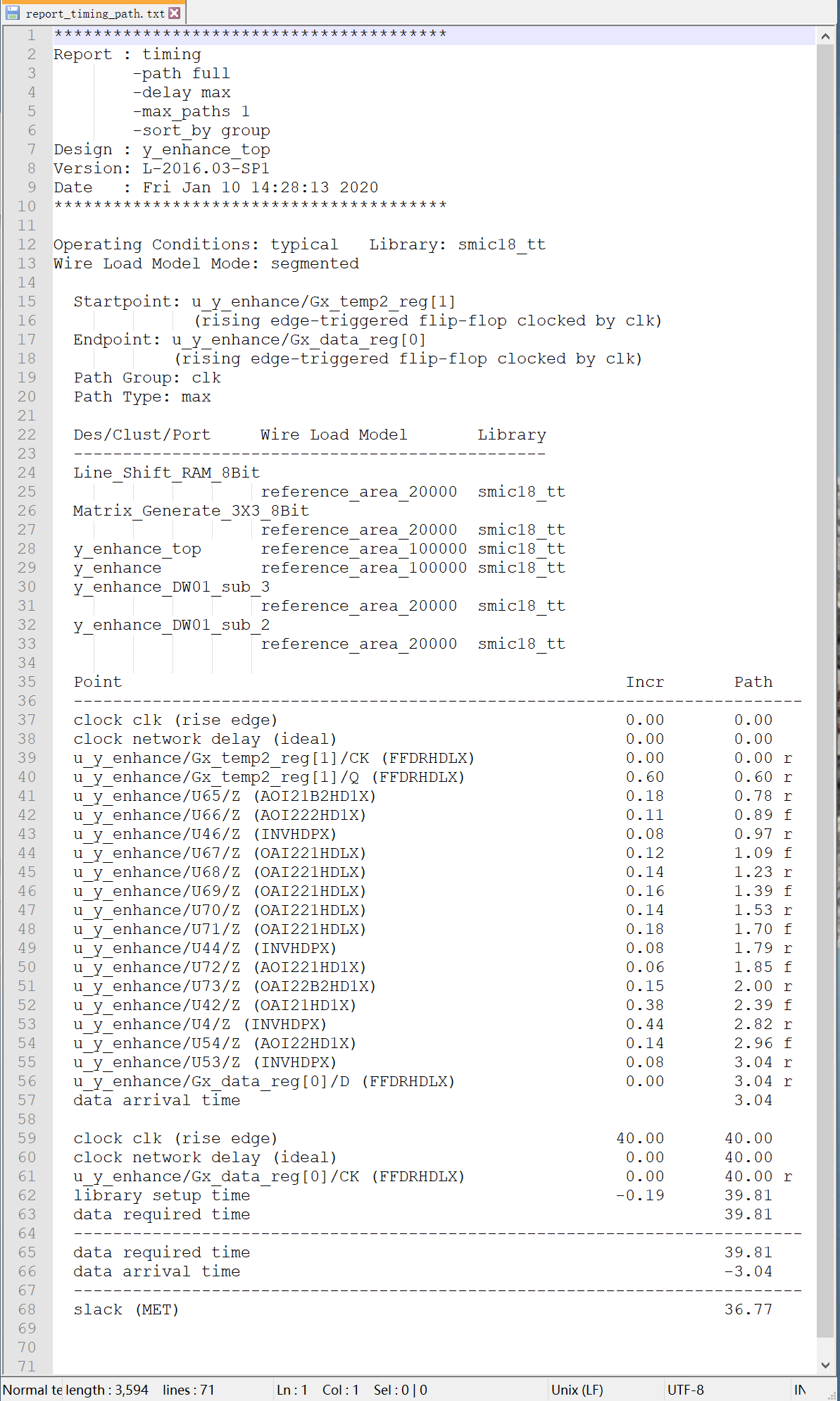
（3）添加约束，其中包括以下几个方面：设置输出负载，设置设计规则约束，设置优化约束，设置时钟；

（4）编译；

（5）生成时序和面积报告，sdf延时文件，保存网表文件；

### 2.2 综合结果

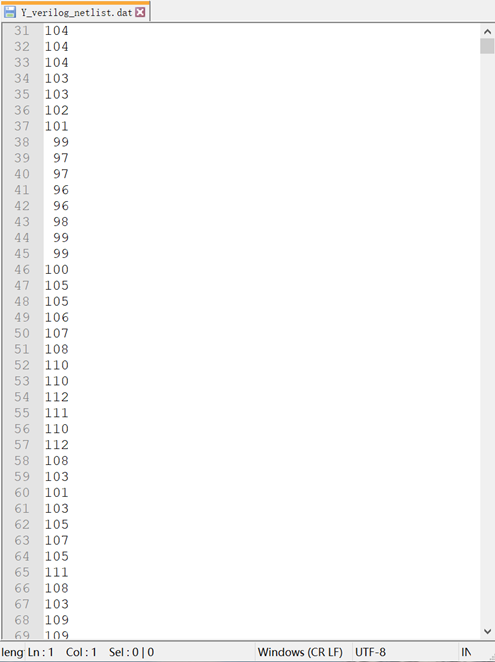
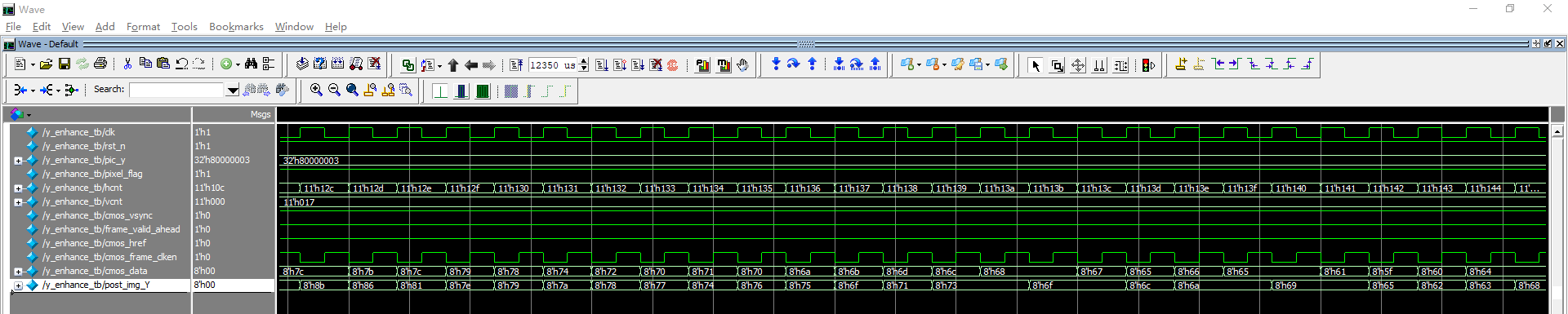
（1）综合生成的面积报告如下图所示：

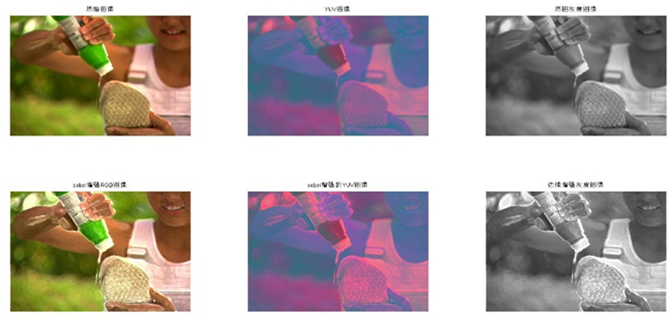
1. 综合生成的时序报告如下图所示：

### 2.3 综合后仿真

（1）将DC综合所用的工艺库文件以及综合生成的网表文件导入Modelsim工具；

（2）在Modelsim工具中，将DC综合产生的时序文件SDF文件反标到网表文件进行仿真，得到输出结果文件和波形如下所示:

（3）将输出增强后的Y数据存储，调用Matlab程序生成RGB图像，同步骤一、步骤二的结果值进行比较，肉眼基本上看不出来差距，可以认为效果是一致的。



## 3. 基于AHB总线的IP核开发及仿真

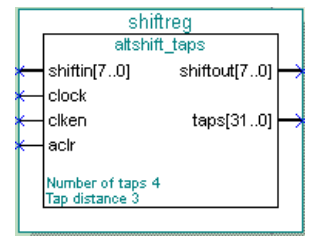
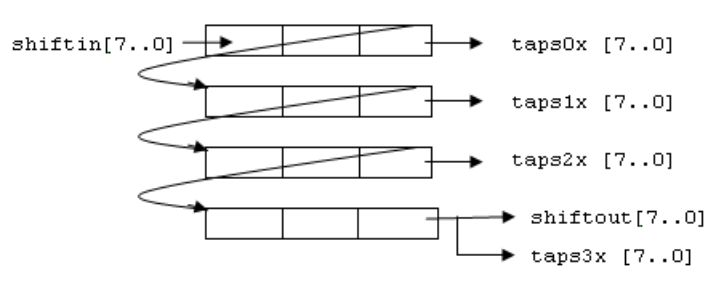
该IP核作为AHB总线的一个Slave，编写测试模块Master，对其进行读写测试。由Master读取得到测试向量，并通过AHB总线传输给Slave进行边缘增强，结果存到Memory中以便后续视频处理及显示之用。AHB总线的仿真结果与行为结果比较如下图所示：

AHB总线仿真 行为级仿真

将灰度图像增强结果同步骤一、步骤二的结果进行比较，差异不大，基本上可以认为效果是一致的。

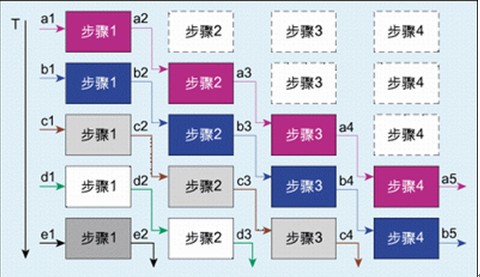
## 4. FPGA验证与仿真

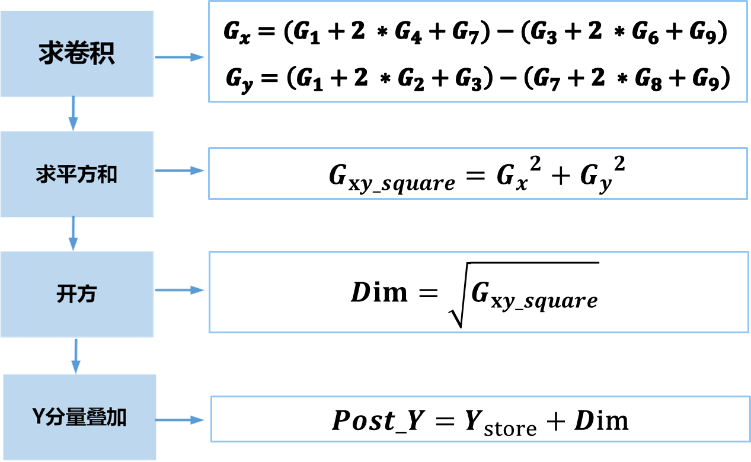
之前编写的Verilog代码程序，没有考虑到在FPGA中运行的实际问题。考虑到FPGA的特点，认真分析Sobel边缘增强算法，对程序进行了改进和优化。

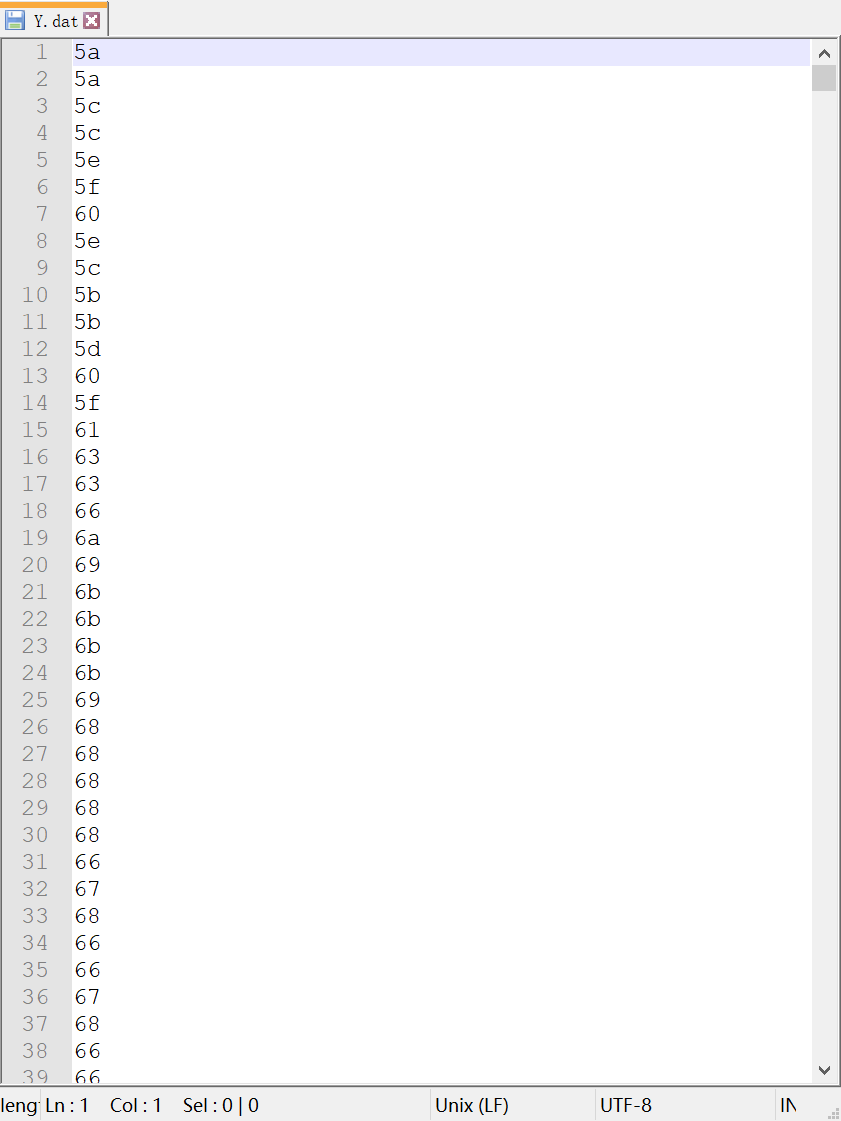
如果一次性读入整张图像，会占用大量逻辑资源。经过分析，最理想的情况下，只需要存储两行图像，当第三行图像输入的时候，就可以进行运算。综合考虑后，程序采用移位RAM寄存器IP核，原理如下图：

由于算法需要进行平方和开方，比较耗费时间，考虑到FPGA计算能力的限制，可能一个周期内无法完成所有运算，导致出错。经过分析，决定采用流水线技术，将整个算法过程分为四步，并行处理。

流水线技术是将组合逻辑系统地分割，并在各个部分之间插入寄存器，并暂存中间数据的方法。目的是将一个大操作分解成若干的小操作，每一步小操作的时间较小，所以能提高频率，各小操作能并行执行，所以能提高提高处理速度，流程如图所示：



程序中，将Sobel边缘增强算法分解为四个过程，依次为求卷积、求平方和、开方、Y分量叠加，流程如图所示：

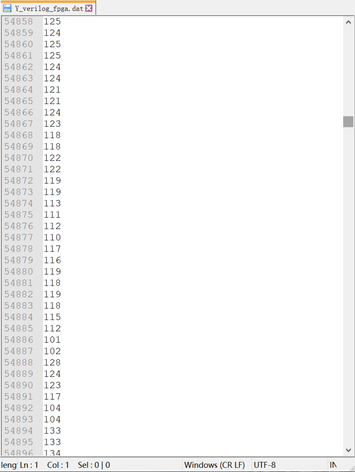
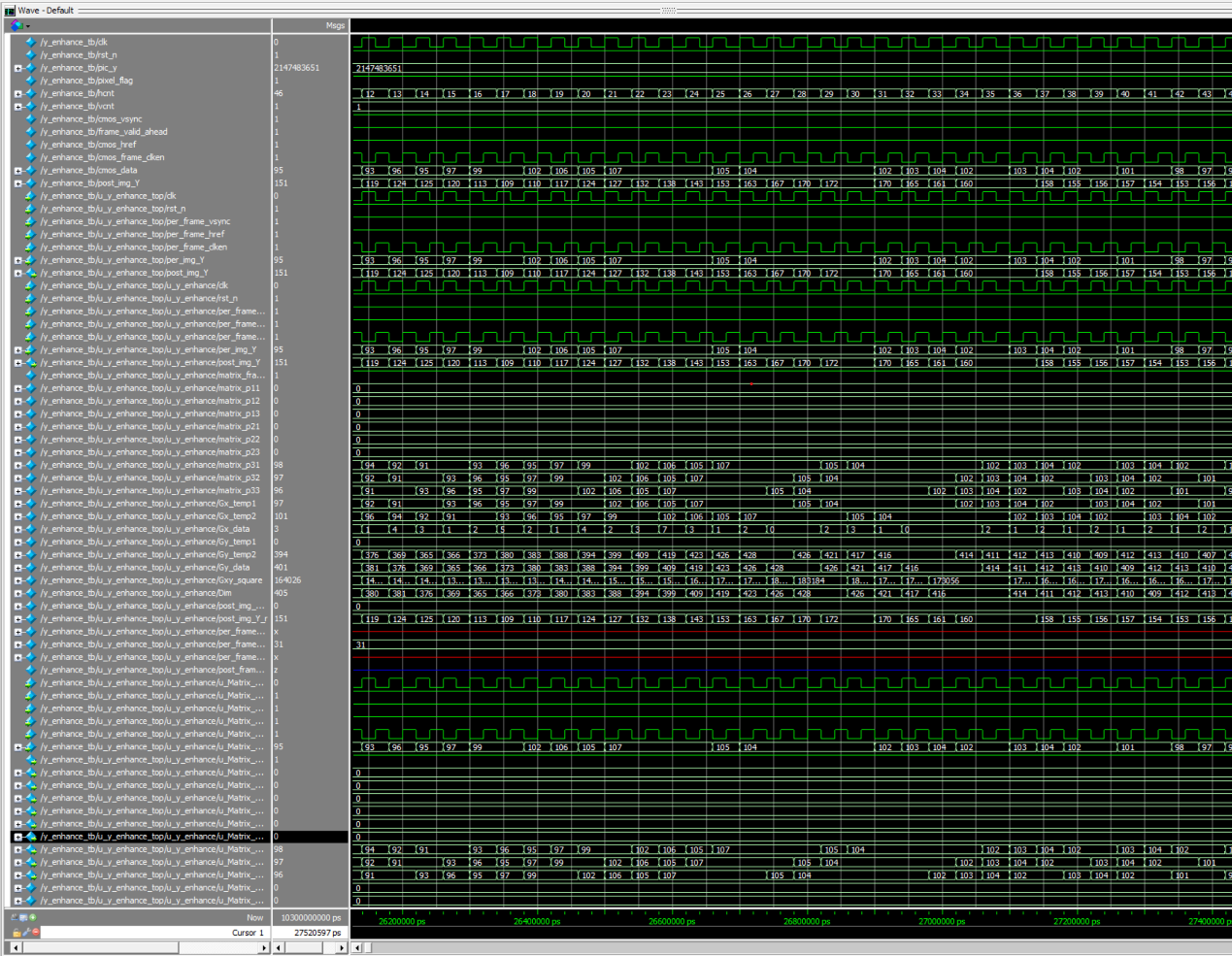
其中，输入的灰度图像数据如下图所示：

具体操作流程如下：

（1）建立工程，选取器件型号为Stratix EP1S25；

（2）将Verilog设计代码y\_enhance.v导入Quatrus II中，编译，生成网表文件和标准延时文件；

（3）配置仿真环境，使用生成的网表文件和时序文件，结合FPGA的库单元进行后仿真，从数据文件Y.dat中读入图像灰度数据，输出增强后的灰度数据并存储。

得到处理后的灰度图像数据和仿真波形如下所示：

（d）调用Matlab程序生成RGB图像，结果如下图所示。

结果同步骤一、步骤二、步骤三的结果进行比较，肉眼基本上看不出来差别，可以认为增强效果是完全一致的。

