E111064503 通訊所 吳紹齊HW3

<https://github.com/cowboy35927/ESL/tree/main/Hw3>

**1.Base Implementation**

這裡是定義通道的header，如果沒有定義就自動定義p2p，反之fifo。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

這裡是filter.cpp，下圖是進行初始化，設定rst和clk。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

下圖是do\_filter進行reset。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

下圖是從testbench去讀取pixel，pixel會經過p2p的channel(i\_rgb)來得到，然後將pixel拆成r、g、b三個部分，分別寫入val[0]、val[1]、val[2]。

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

然後進行Median Filter，取出Median pixel。

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

下圖是Mean filter，將得到的median pixel放入buffer後經過Mean filter，得到mean pixel，然後再分別乘2^16、2^8、1，放入total，最後經過p2p channel(o\_result)到testbench。

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

下圖是testbench，定義通道的header，如果沒有定義就自動定義p2p，反之fifo。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

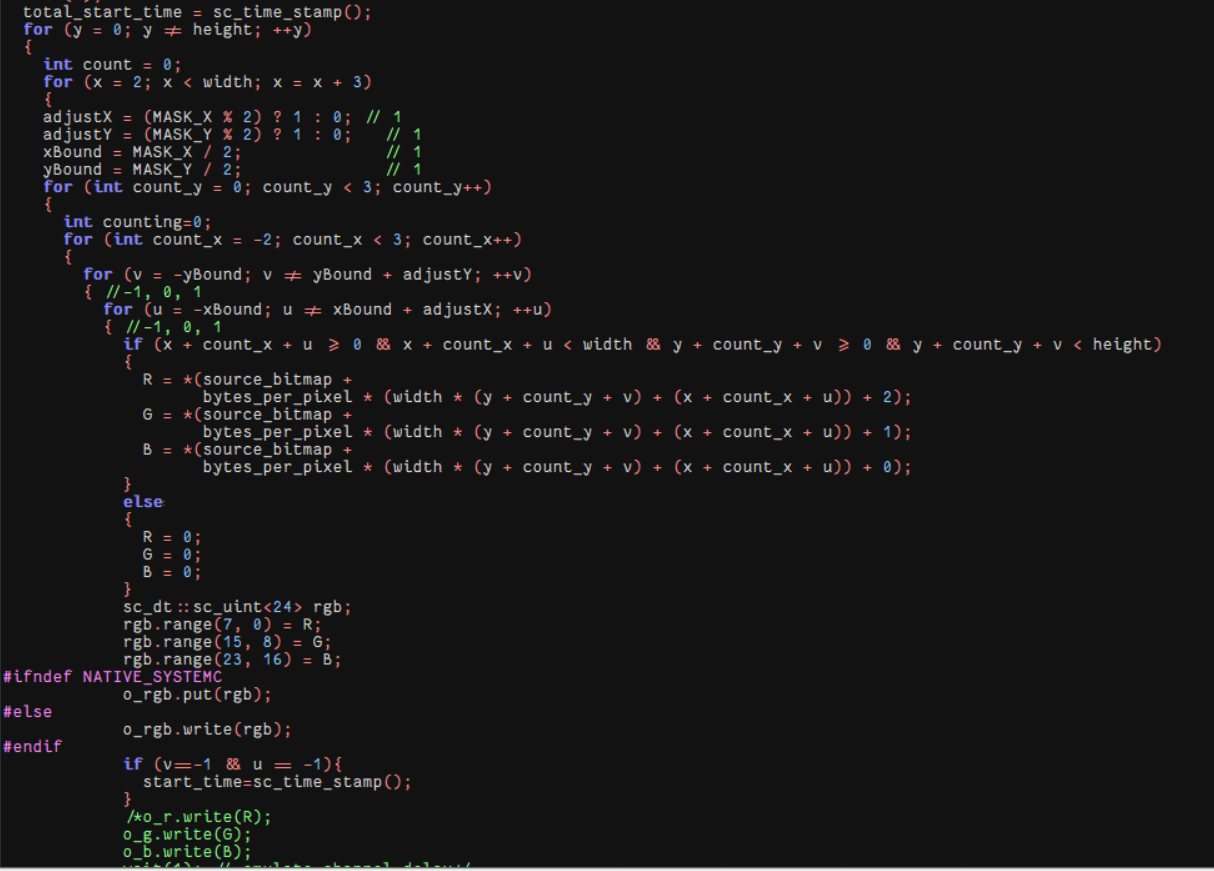
自動產生的描述

下圖是testbench進行初始化，設定clk。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

下圖是testbench的feed\_rgb，是將讀進來的圖片去解析出每個pixel的rgb，將解析出來的rgb經過p2p channel(o\_rgb)輸出到filter，latency在讀第一個pixel後寫入當時時間start\_time，為了算latency。

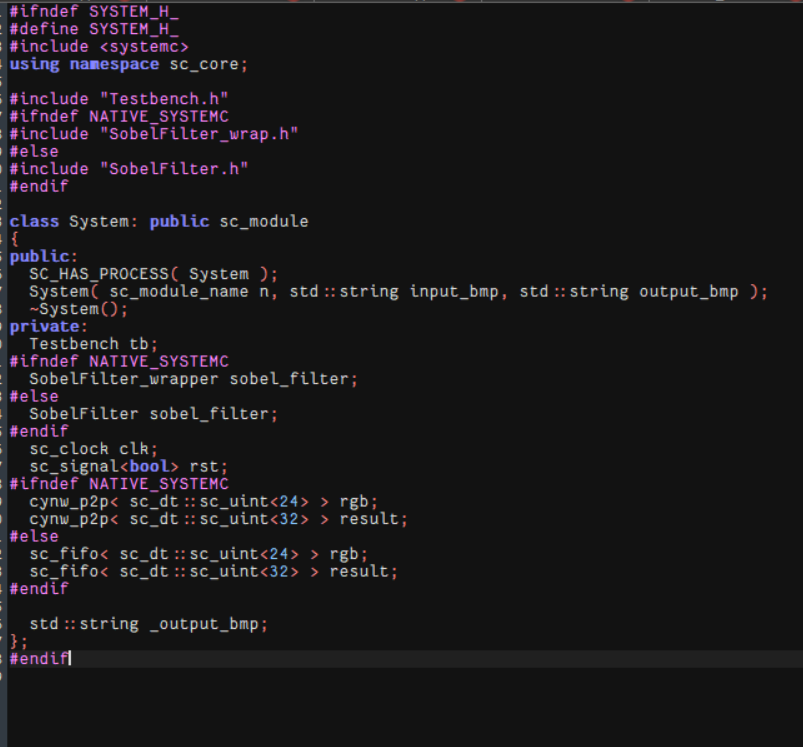


下圖是fetch\_result，將得到的pixel寫成bmp的圖片，會從p2p channel(i\_result)得到pixel，再去解析rgb後分別寫入。

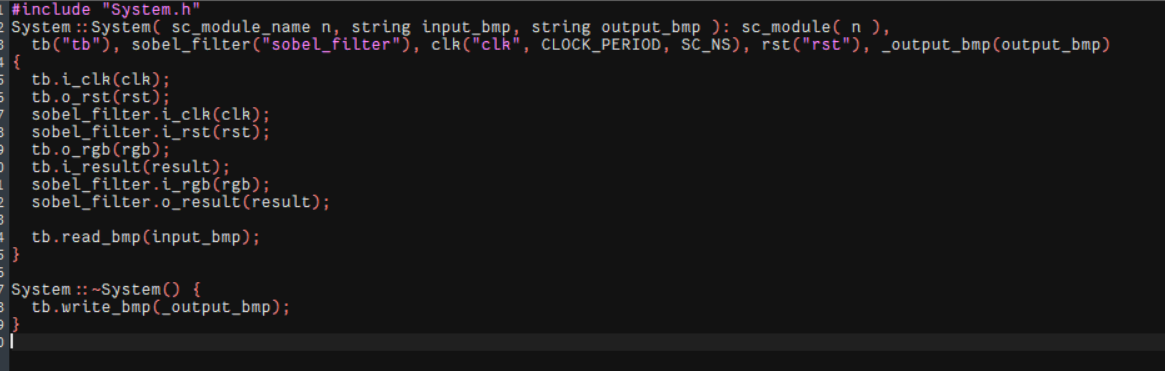
一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

下圖是system.h，用來定義通道的位數和合成模塊。

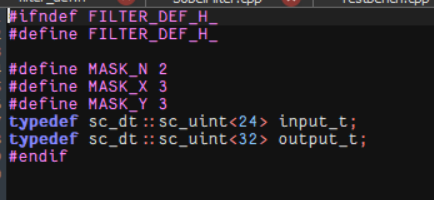


這裡是systempipeline.cpp，類似於verilog的top模塊，用於連接所有模塊，從testbench通過rgb通道進行filter，再通過result通道返回testbench，最後輸出bmp。



**2.Improve coding styles**

**使用位寬來約束運算符。**



**在filter.cpp裡將rgb轉換和mean filter的乘法改成shift。**

一張含有 文字, 螢幕擷取畫面, 陳列, 字型 的圖片

自動產生的描述

**將testbench的除法改成shift。**

一張含有 螢幕擷取畫面, 文字, 字型, 行 的圖片

自動產生的描述

**3. Optimized Implementation**

**在do\_filter裡的while前加上while\_1。**

一張含有 文字, 電子產品, 螢幕擷取畫面, 陳列 的圖片

自動產生的描述

**下圖是HLS configuration，多加了unroll\_loops和constrain\_latency的指令**

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

**下圖是simulation 對V\_UNROLL\_ALL。**

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

**比較(BASIC):**

**1.256\*256大小的圖片:**

1. **Base Implementation:**

Area:

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Run time: 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Latency:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

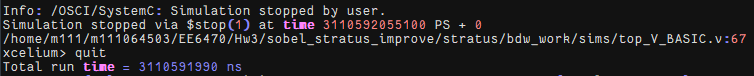
Throughtput=281,162 pxiel/s。

(2) **Improve coding styles**

Area:

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Run time: 

Latency:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Throughtput=281,162 pxiel/s。

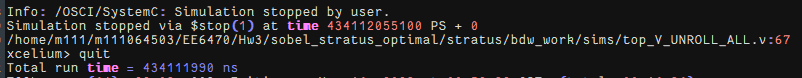
將乘法改成shift後，優化過的code合出來的area、latency、run time、throughput都一樣，我認為應該是tool已經能自己優化乘除法器了，所以原本是用\*/的部分和shift合出來的結果會一樣。

(3) **Optimized Implementation**

Area:

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Run time: 

Latency:

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Throughtput=1,886,792 pxiel/s。

使用unroll後，面積暴增到了4萬，因為將loop迴圈展開，所以面積會比原本多了6倍多，但latency也少了6倍多，run time則是快了7倍多，throughtput則多了6倍多，所以這邊是以面積換取速度。

**2.512\*512大小的圖片:**

1. **Base Implementation:**

Area:

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Run time: 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Latency:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Throughtput=281,162 pxiel/s。

(2) **Improve coding styles**

Area:

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Run time: 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Latency:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Throughtput=281,162 pxiel/s。

將乘法改成shift後，優化過的code合出來的area、latency、run time、throughput都一樣，我認為應該是tool已經能自己優化乘除法器了，所以原本是用\*/的部分和shift合出來的結果會一樣。

(3) **Optimized Implementation**

Area:

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Run time: 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Latency:

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Throughtput=1,886,792 pxiel/s。

使用unroll後，面積暴增到了4萬，因為將loop迴圈展開，所以面積會比原本多了6倍多，但latency也少了6倍多，run time則是快了7倍多，throughtput則多了6倍多，所以這邊是以面積換取速度。

一張含有 人的臉孔, 服裝, 時尚配件, 人員 的圖片

自動產生的描述一張含有 服裝, 人的臉孔, 頭飾, 時尚配件 的圖片

自動產生的描述

一張含有 甜椒, 蔬菜, 生產, 天然食品 的圖片

自動產生的描述一張含有 蔬菜, 甜椒, 生產, 天然食品 的圖片

自動產生的描述