**MSoC Self-paced 1: Xdesign dataflow\_stable\_content**

**R08943011 黃文璁**

* **[v] HLS C-sim/synthesis/cosim**
* **[v] System bring-up (zedboard)**
* **[v] Improvement (latency, area)**
* **[v] Github:** [**https://github.com/b04901060/MSoC-Application-Acceleration-with-High-Level-Synthesis**](https://github.com/b04901060/MSoC-Application-Acceleration-with-High-Level-Synthesis)

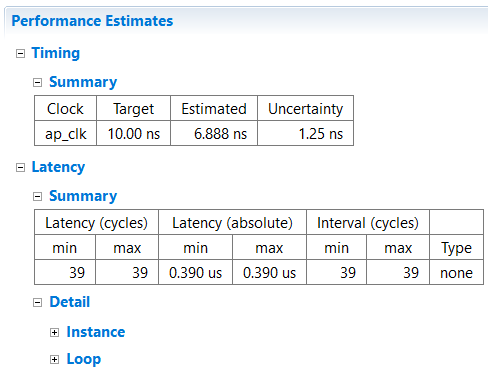
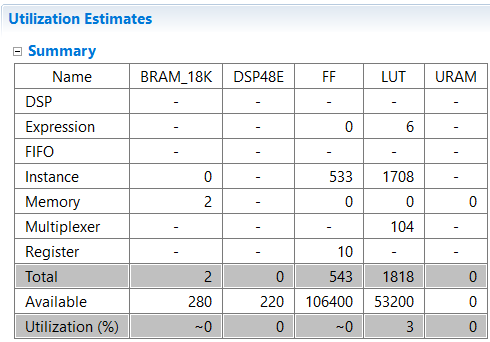
1. **Introduction**

本example主要用來展示如何使用 #pragma HLS stable variable=X 指令來消除使用 #pragma HLS dataflow時產生的額外同步操作。

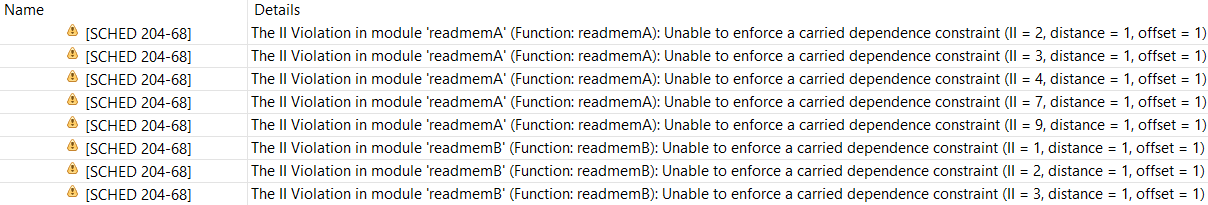
這個 example中實作功能如下：

給定陣列A和B，計算A[B[i]] 且 B[i]!=0 的總和。

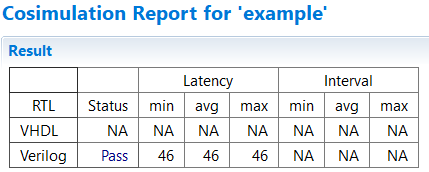
1. **C Synthesis and cosim**

** **

另外可以注意到合成結果出現II violation：



**Cosim:**

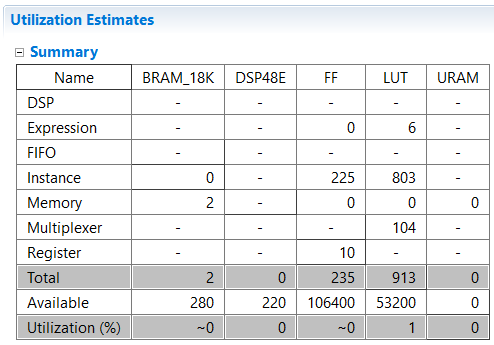
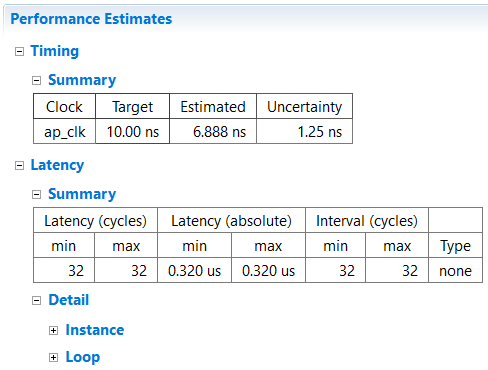


**-> Verilog pass**

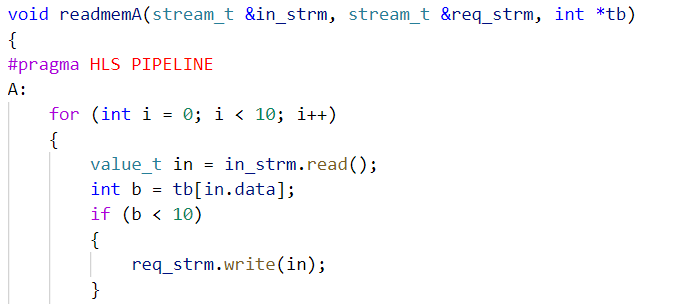
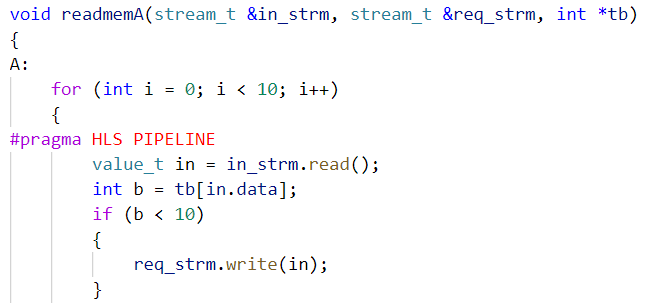
1. **Optimization**

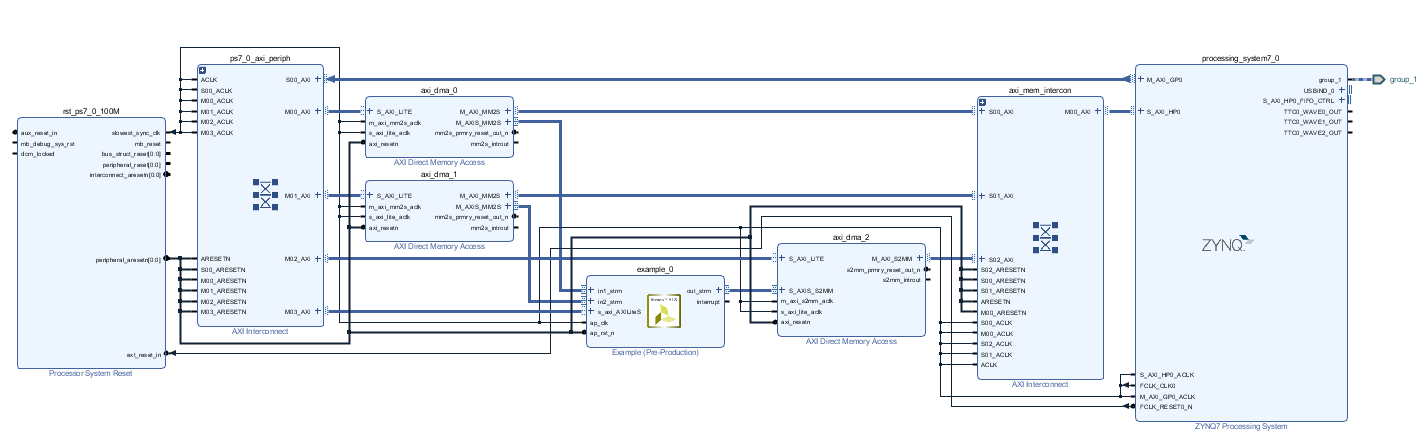
注意到原本的 code出現II violation使得II無法為1。將原本放在function top的 #pragma HLS PIPELINE 移動到for loop內部後，合成後便可達到II=1。但實際上整個函數只有一個 for loop，在function top和for loop top使用PIPELINE應該要有相同的結果，具體原因還不太確定。

* + **Optimized synthesis result**

****

* + **Latency**: 39 -> 32 (**-18%**)
  + **FF**: 543 -> 235 (**-57%**)
  + **LUT**: 1818 -> 913 (**-50%**)

**Before After**

1. **Block Diagram**

系統整合完成後，便能使用和lab2類似的方式，以python程式操作DMA和kernel間的資料搬動：

