Multimedia SoC Design

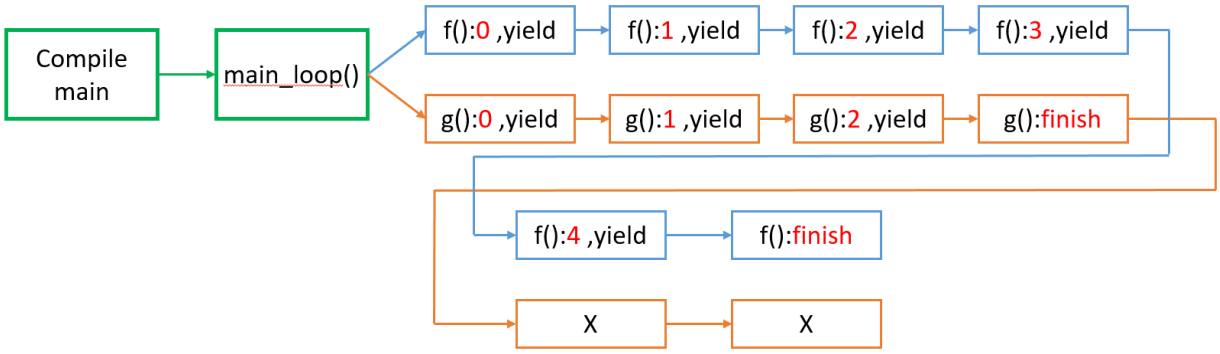
Homework #2 Report

R06943087 葉玲瑜

Example #0

(1) Draw a complete program execution flow

Ans:



p.s藍色的block(f())會先執行，然後橘色block(g())才執行

(2) Explain how the zip_iterator works

Ans:

zip_iterator是一種迭代器，像是在example#0中zip_longest(f(),g())這個迭代器裡面，f()和g()都是都是迭代器的子集合，所以他們會一起被迭代；每個iteration，子集合(在這裡是f()和g())都會輸出一個元素，一直到擁有比較多元素的那個子集合(f())把所有元素都輸出完畢為止；所以在example#0裡，f()的元素比較多，所以最後兩個iteration只有f()輸出而g()早已迭代完畢。

Example #1

(1) What if we use Comsumer(11)?

Ans:

程式會陷入無窮迴圈，因為Producer只會產生10個，但是Comsumer卻要拿到11個item，所以條件永遠無法滿足就會一直卡住。

(2) What if we swap the order in main_loop()?

Is there any difference? Do you think this is reasonable?

Ans:

會有不同，因為一開始先執行的變成是Comsumer，而Producer還沒有開始產生item，這樣會要多一個clock。

Example #3

(1) What does dont_initizlize mean in SystemC

Ans:

在SystemC裡面，是用SC_METHOD來達成dont_initialize的功能，讓程式不要在initial的時候先執行一遍，而是執行到它的時候才做動作。

(2) How do you implement it?

Ans:

我把一開始的clock function那邊改掉，然後不讓event_pending去拿INIT_EVENT的值。

(3) Give an example about the difference with/without dont_initialize

Ans:

像這份example#2_2沒有做dont_initialize的話，就會先印出handing 5，因為在initial的時候，程式會先執行一次，這時候init_event就會被印出來；所以如果要避免掉先被執行的情況，這時候就要設一個dont_initialize，確保程式不會在吃到Producer產生item前的值。