Computer-Aided Design for VLSI Design

Homework4（Student ID: ntust_B10832019｜Name: 林琛琛）

1. Provide a simple explanation of your code.

亂數產生器(RNG)的程式，包含：CA、RNG 跟 test_bench 是直接使用上課講義的程式。

加分題的樂透數字程式，要先啟用 RNG 這個之前就寫好的 component：

```
random : RNG port map(seed, clk, sel, rng_out);
```

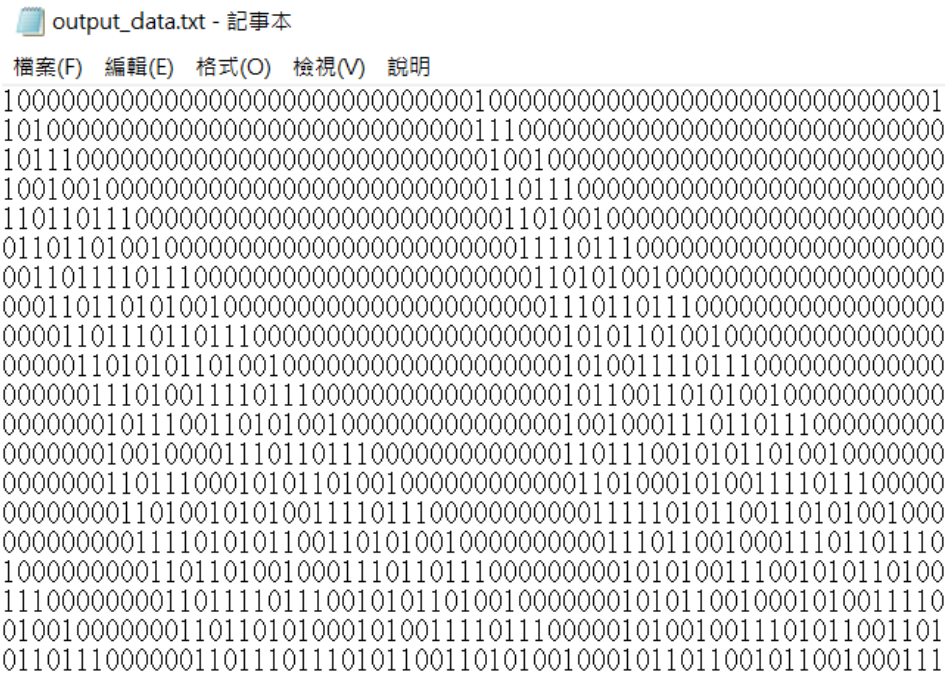
以及把產生的亂數求餘數：

```
module_out <= to_integer(unsigned(rng_out)) mod 21;
```

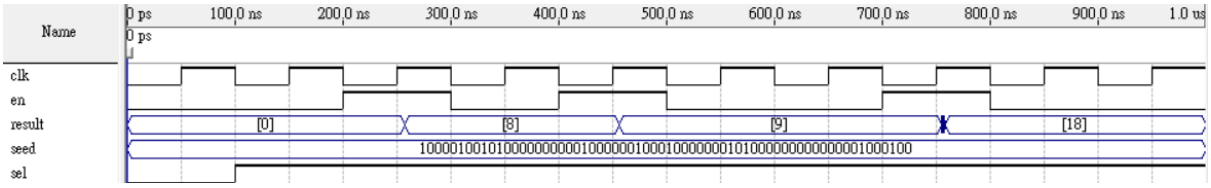
最後，確保當 rising edge 並且 en = '1' 的時候更新輸出：

```
process(clk)
begin
	if clk'event and clk = '1' then
		if en = '1' then
			result <= module_out;
		end if;
	end if;
end process;
```

2. Waveform diagram here (Simulation Results)



加分題：



產生了三個亂數，分別是 8、9、18。

3. Reflections and discussions

這次的作業讓我熟悉如何撰寫 test bench，以及基礎的 modelsim 操作。在產生 64 bit 的亂數方面沒有遇到什麼問題，基本上照著講義的步驟操作即可。

加分題的部分，第一個卡關的是 mod。RNG 的 output 是 64 bit 的 vector，但 mod operator 不接受這個資料型態，必須做轉換。
當時的第一個想法是寫一個 function vector_to_int，但這個想法行不通，因為超出 integer 能儲存的範圍上限。後來查文件的範例，才發現有更簡便的方法完成取餘數的動作。

遇到的另一個困難是在波型模擬時，剛開始的 seed 偷懶只把最後一個 bit 設 1，其餘保持零，結果模擬出來的波型以 period = 100 ns 的 clk 為例，大概要 600 ns 以後才開始有亂數產生，前面時候的 result 都是 0。當時檢查很久，不管是程式碼還是波型圖，都找不出問題所在，後來猜測是 seed 設得不好。之後把 seed 設亂一點後就能正常產生亂數了。

我推測原因是初始的 1 太少時，要經過比較多次的亂數產生，才能有一個夠「亂」的數字。最開始的 seed 前期可能恰好亂數都太小，mod 出來都是 0，也就看不出變化。

此外，由上面產生的 64 bit 亂數產生列表以及課程講義上面的圖片，可以觀察到隨著一次次產出新的亂數，數字 1 會逐漸蔓延開來，直到整個字串亂到沒辦法辨別變化。