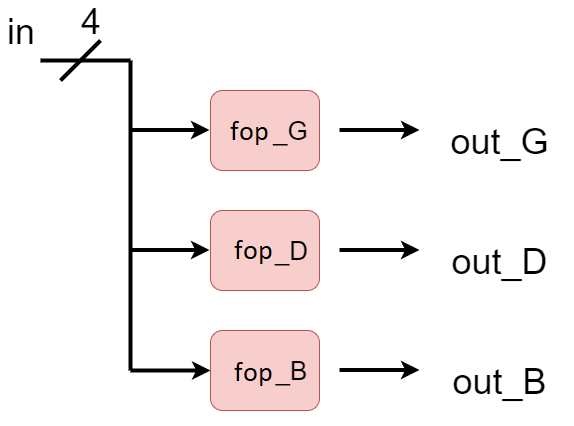
Lab1: Fibonacci or Prime number detector

Output 1 if input is a Fibonacci number or a prime number. Otherwise, output 0.

Input range= [0,15] (Use 4 bit to represent the input)

1. Concept



1. Truth table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input | in[3] | in[2] | in[1] | in[0] | out |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 0 |
|  |  |  |  |  |  |

1. K-map (a=in[3], b=in[2], c=in[1], d=in[0])

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ab cd | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 0 | 1 |
| 01 | 1 | 1 | 1 | 0 |
| 11 | 1 | 1 | 0 | 1 |
| 10 | 1 | 0 | 0 | 0 |

PI: (~b)(~c)(~d), b(~c)d, (~b)cd, (~a)(~b), (~a)d

EPI: (~b)(~c)(~d), b(~c)d, (~b)cd, (~a)(~b), (~a)d

out = (~b)(~c)(~d) + b(~c)d + (~b)cd + (~a)(~b) + (~a)d

*\*不知道這樣的寫法是否會有hazard 問題？例如7->8沒有共同的圈圈。*

1. Circuit design

or\_0

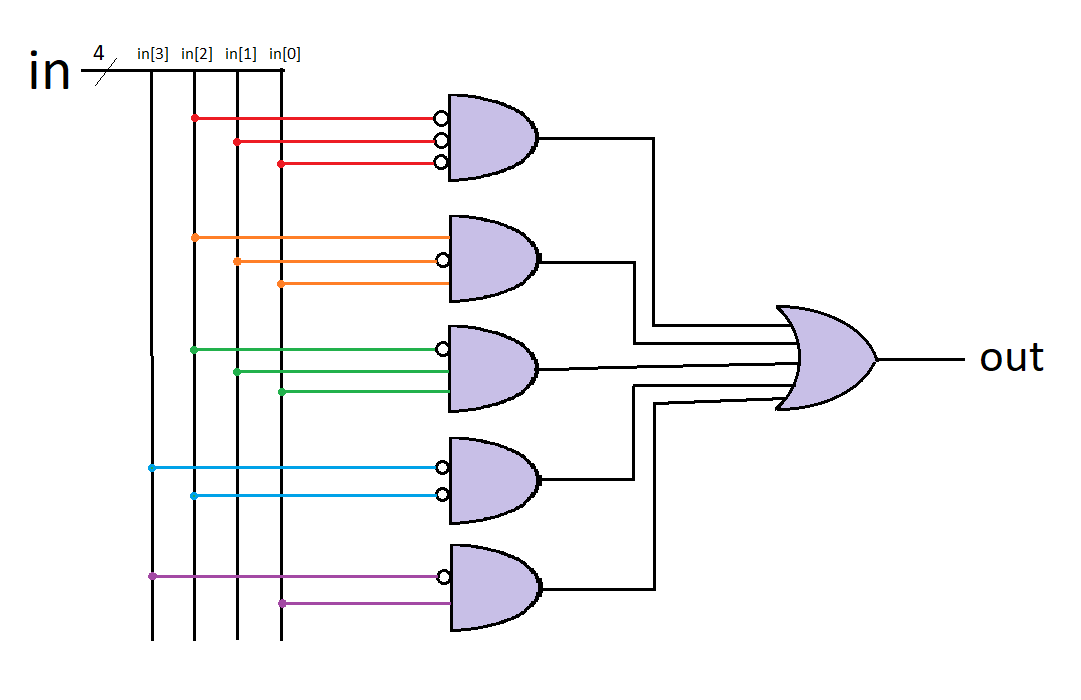
and\_1

and\_0

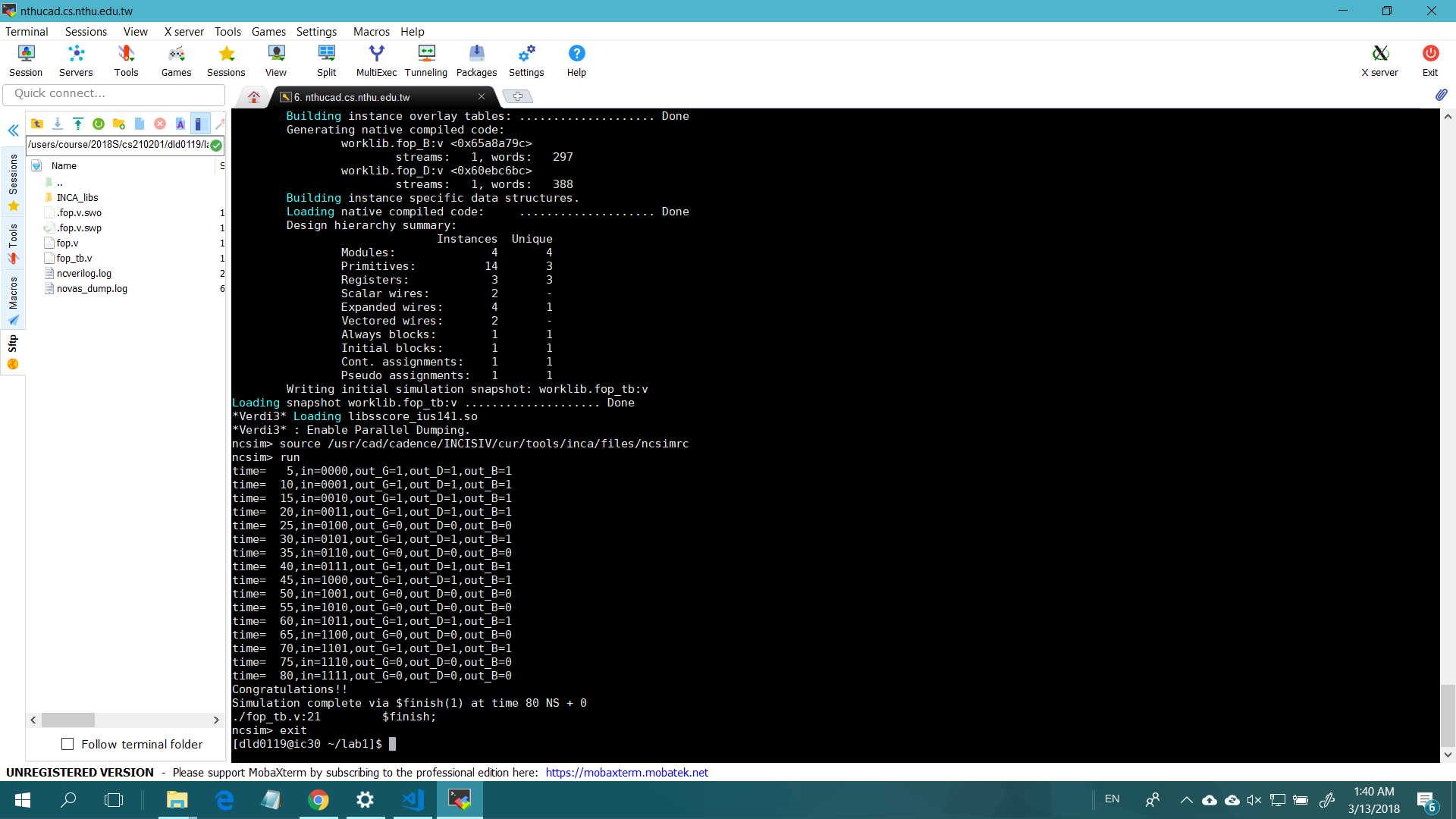
and\_2

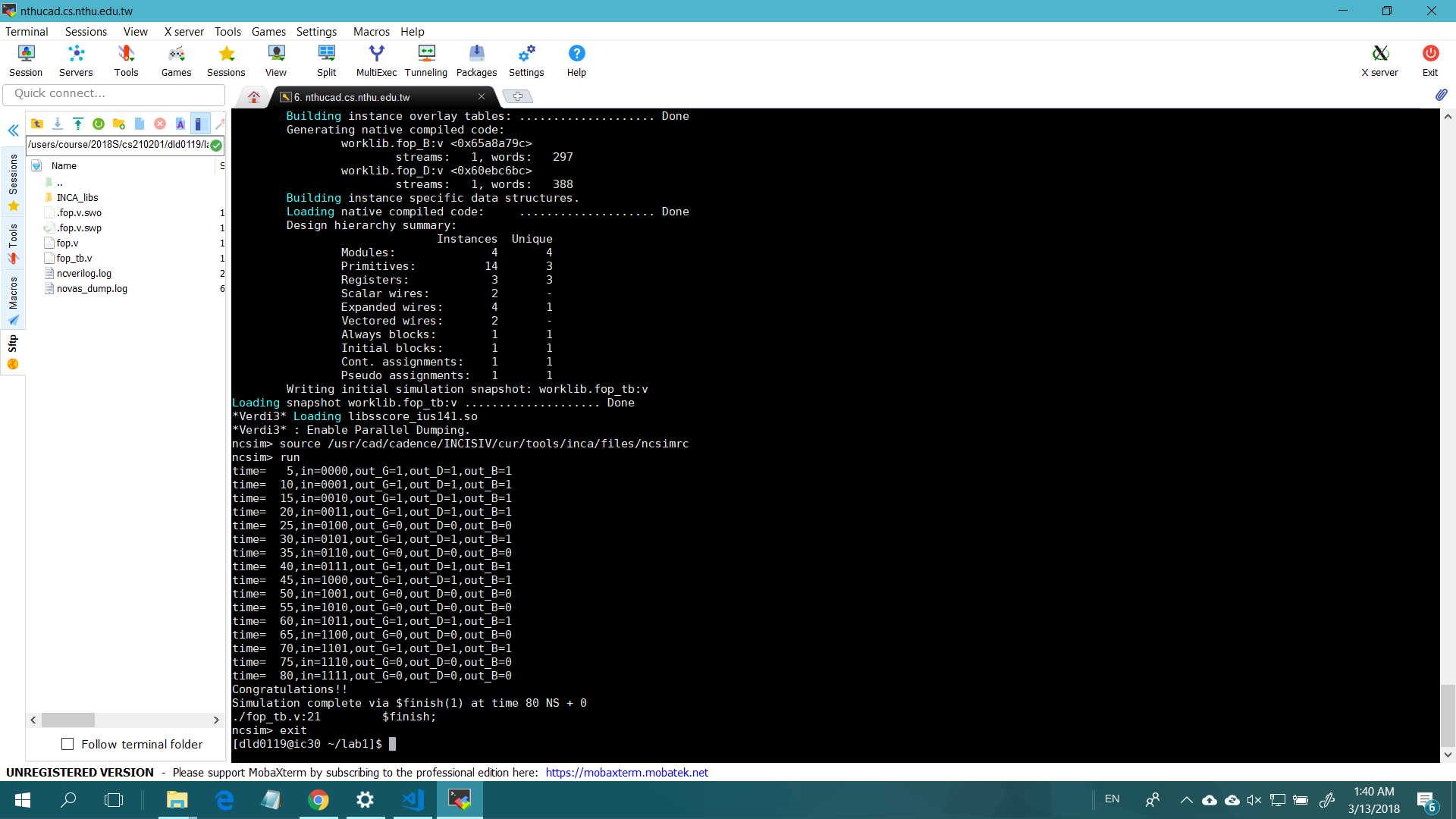
and\_3

and\_4



1. Simulation Result





1. Discussion

課程網頁上的介紹十分詳細，很多我們內心的疑問例如wire和reg、i++和i+1等在上面都有解答，令我們茅塞頓開。可是我還是有一些疑問：

1. 為什麼0被包含在fop裡面？據我所知0不是prime number也不是fibonacci……
2. **如何考慮hazard問題？**

是只要沒有size為1的圈圈就可以了嗎？還是看兩個連續而且都為1的數字有沒有被同一個圈圈起來？像這題的K-map中的7（0111）和8（1000）並沒有被圈在同一個圈圈裡面，因此想問助教這樣會出現hazard嗎？老師有解釋過hazard的原因，但是我還是不知道如何看出要在哪些地方加圈圈來避免hazard。

1. **在不同的description要用不同的數字表示法？**

為什麼在Behaviour Description的always block中可以直接以數字0, 1, 2, 3, 5, 7, 8, 11, 13來表示，而不需分開4個input來表示（像Dataflow Description）？  
而為什麼out要寫成1'b1、1'b0，in可以寫成0, 1, 2, 3, 5, 7, 8, 11, 13不需要寫bit和表示法？是因為verilog會自動理解in嗎？

1. **關於程序碼執行順序的問題**

課程網頁上有說硬體語言的程式不是一行行執行的。

但是always block裡面寫的程式有開始跟結束，是“動作”，所以是從上到下一行行執行的。而其它的描述就像是在描述一個圖片/結構而不是描述從頭到尾的一串事件/動作，所以沒有一行行先後順序的差別。請問這樣的理解對嗎？

1. **Testbench**

課程網頁上的testbench寫法和範例程式的testbench寫法似乎有很大的不同，感覺範例程式的testbench寫法比較好懂，前者幾乎看不太懂……

fop\_G hvg

(

.in(in),

.out(out\_G)

);

想問fop\_G後面的hvg是什麼意思啊？是幫呼叫取名字嗎？

1. **Vim很難用**

一般都是在外面寫完再copy進去vim，但是要修改code很難，用gg（cursor到第一行）然後dG（清除後面全部）把code清掉後再重新貼上新的code，有時會出現舊的code參雜在裡面的情況。