

邏輯設計實驗 Lab1 結報

105060012 張育菘

1. Design and implement a full adder. ($s + \text{cout} = x + y + \text{cin}$)

1.1 Write the logic equation.

1.2 Draw the related logic diagram.

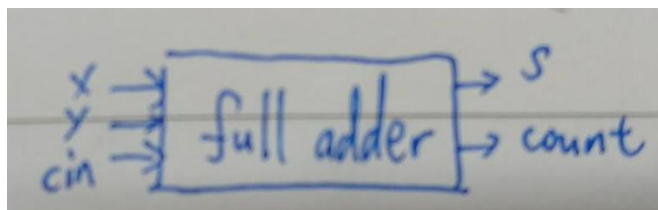
1.3 Verilog RTL representation with verification.

Design Specification

input : x, y, cin;

output : count, s;

block diagram :



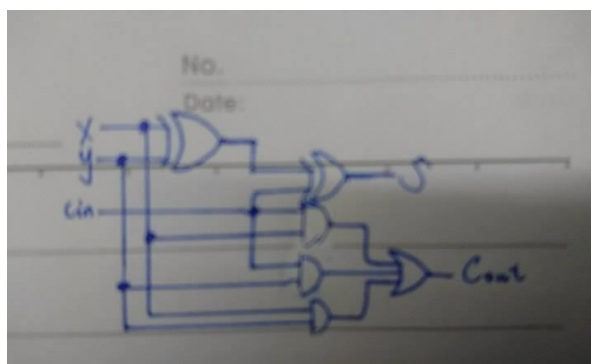
Design Implementation

Logic function :

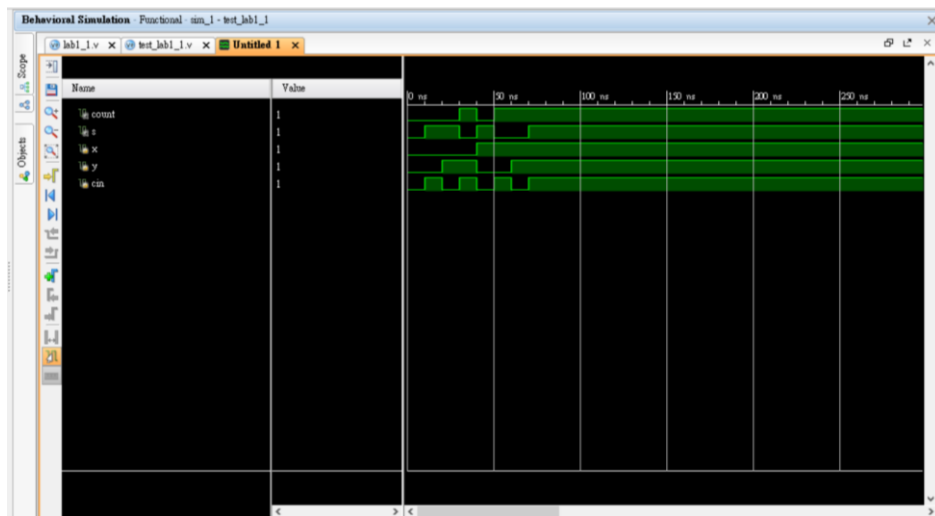
$$s = x \oplus y \oplus \text{cin}$$

$$\text{count} = (x \& y) \mid (x \& \text{cin}) \mid (y \& \text{cin})$$

Logic diagram :



Result :



Discussion

1. $(x \& (\sim y) \& (\sim cin)) \mid ((\sim x) \& (\sim y) \& cin) \mid (x \& y \& cin) \mid ((\sim x) \& y \& (\sim cin))$ 可以簡化成 $x \wedge y \wedge cin$

2. Design a single digit decimal adder with input A(a3a2a1a0), B(b3b2b1b0), Cin(ci), and output S(s3s2s1s0) and Cout(co).

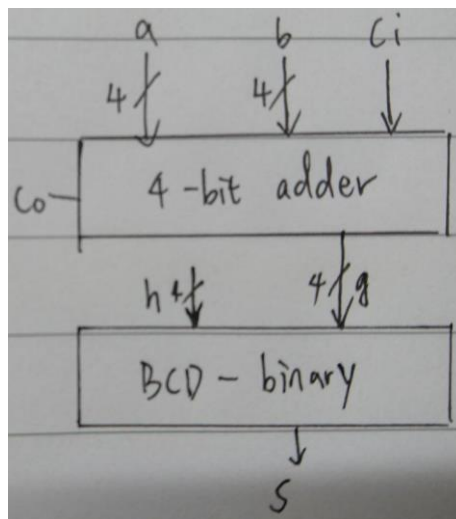
Design Specification

input : [3:0]a, [3:0]b,ci;

output : co, [3:0]s;

wire [4:0]f,[3:0]g,[3:0]h;

block diagram :



Design Implementation

Logic function :

$$f = a + b + ci;$$

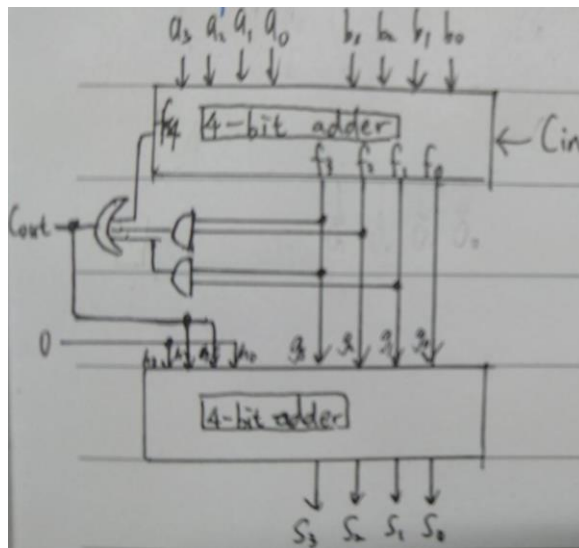
```
co = f[4] | (f[3]&f[2]) | (f[3]&f[1]);
```

```
g[0] = f[0]; g[1] = f[1]; g[2] = f[2]; g[3] = f[3];
```

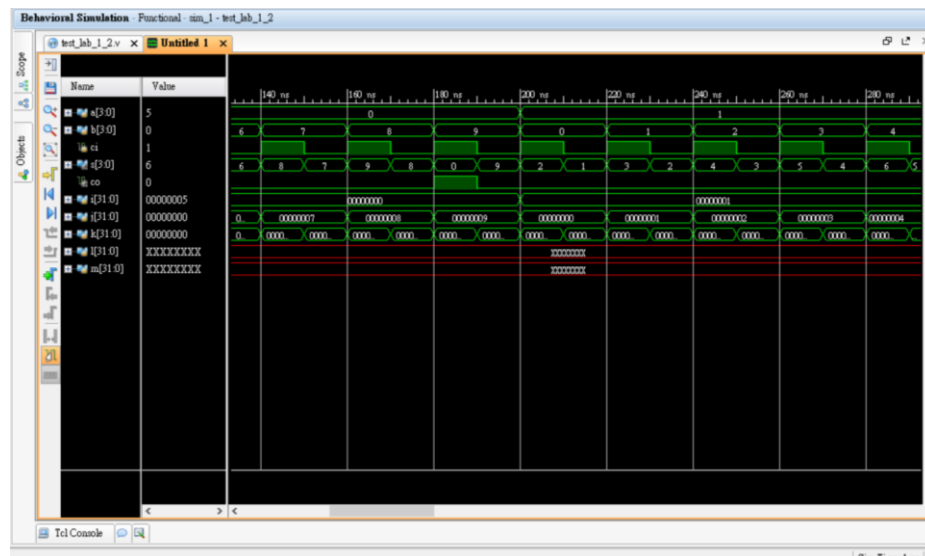
```
h[0] = 0; h[1] = co; h[2] = co; h[3] = 0;
```

```
s = g+h;
```

Logic diagram :



Result :



Discussion

1. 在打 verilog 時，a、b 與 f 為多 bits，" $f = a + b + ci$ "這種形式是可以存在的，若有進位，程式軟體會自動幫助你進位，不用再詳細寫內部加法的過程，以此即可。

2. 當你覺得你的 code 沒有問題，但是模擬卻一直跑不太出來可能是你的修改次數過多，導致 vivado 讀不太到資料，因此只要重開一份 project 即可。

3 (Bonus) Design a 3-to-8-line decoder with enable (input in[2:0], enable en and output d[7:0]).

3.1 Logic equation,

3.2 Logic schematic,

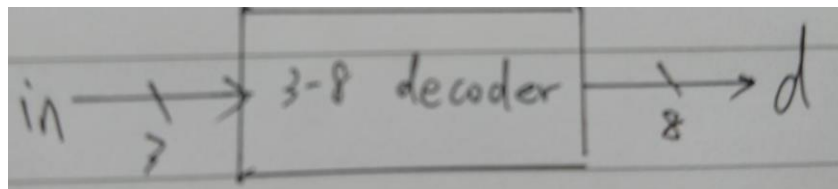
3.3 Verilog RTL representation with verification.

Design Specification

input : [2:0]in,en;

output : [7:0]d;

block diagram :



Design Implementation

Logic function :

$d[0] = en \& (\sim in[0]) \& (\sim in[1]) \& (\sim in[2]);$

$d[1] = en \& (in[0]) \& (\sim in[1]) \& (\sim in[2]);$

$d[2] = en \& (\sim in[0]) \& (in[1]) \& (\sim in[2]);$

$d[3] = en \& (in[0]) \& (in[1]) \& (\sim in[2]);$

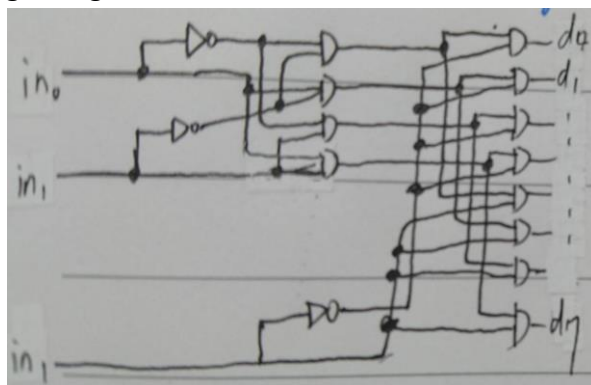
$d[4] = en \& (\sim in[0]) \& (\sim in[1]) \& (in[2]);$

$d[5] = en \& (in[0]) \& (\sim in[1]) \& (in[2]);$

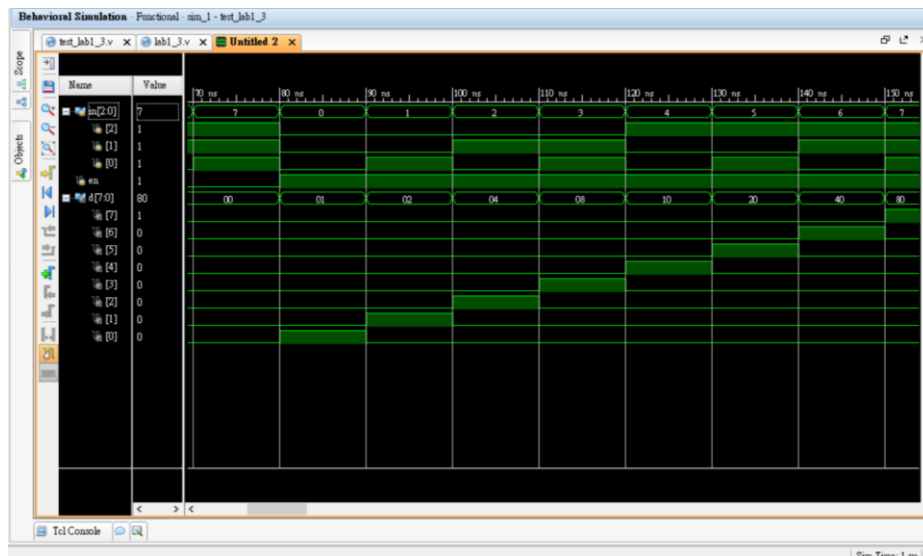
$d[6] = en \& (\sim in[0]) \& (in[1]) \& (in[2]);$

$d[7] = en \& (in[0]) \& (in[1]) \& (in[2]);$

Logic diagram :



Result :



Discussion

1. 當 enable(en)=0 時，output(d)不受 input(in)影響；當 en=1 時，會把 in 進行解碼 in=00→d=0001；in=01→d=0010；in=10→d=0100；in=11→d=1000

Conclusion :

這是繼上學期後再次打 verilog，因為上學期我覺得我沒有學得很好，因此這次我學得更加的認真，而這次打 verilog 也讓我收穫很多，再次習得如何打 verilog，也很開心，能夠把自己做的 logic diagram 打成 verilog，真的是挺開心的。