數位系統導論實驗 LAB06
Verilog Structural Modeling & Timing Simulation

助教 : 徐孟澤, 簡睿宇

Outline

- 實驗目的
- Review : Verilog basics
- RCA w/ Verilog structural modeling
- Gate delay & Timing simulation
- 實驗範例
- 實驗作業
- 評分方式

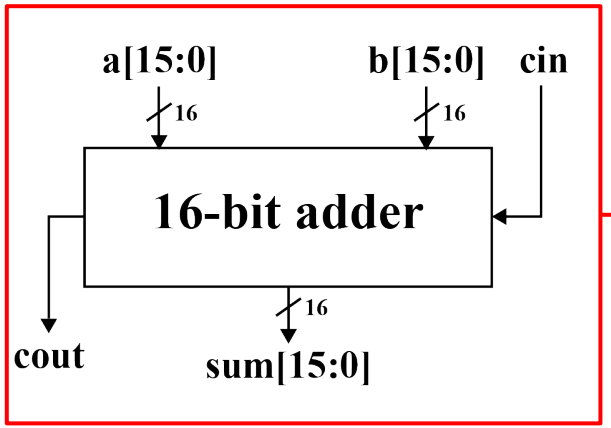
實驗目的

- 在本次的實驗中, 同學們將複習在 LAB 1 學習過的 Verilog behavior modeling & structural modeling 技巧。以邏輯閘層次加法器組成的 ripple carry adder (RCA) 為範例, 以 structural modeling 實作出 16-bit RCA, 最後再利用 LAB 2 學習的 Verilog gtkwave 觀察其 gate delay

Review : Verilog basics

Review : Verilog basics (1/2)

- 下圖為 16-bit adder 與其 Verilog 的行為描述, 藍框為其行為描述區塊, 可以用 behavior modeling 和 structural modeling 方法設計, 以下會對先對 behavior modeling 作介紹




```
module adder_16bit(a,b,sum,cin,cout);

      input [15:0] a,b;
      output [15:0] sum;
      input cin;
      output cout;

      Continuous assignment & Procedural assignment

endmodule
```

Review : Verilog basics (2/2)

- 下列行為區塊為 behavior modeling 的兩種設計方法
  - Continuous assignment : 以 assign 描述硬體架構連結, 位於 always 和 initial 等 procedural block 外, 並且等式左側必為 wire
  - Procedural assignment : 其 behavior code 位於 always 和 initial 等 procedure block 內, 賦值型態為 reg、integer、時間變數...等變數, 但不可為 wire

Continuous assignment

```
assign {cout,sum} = a + b + cin;
```

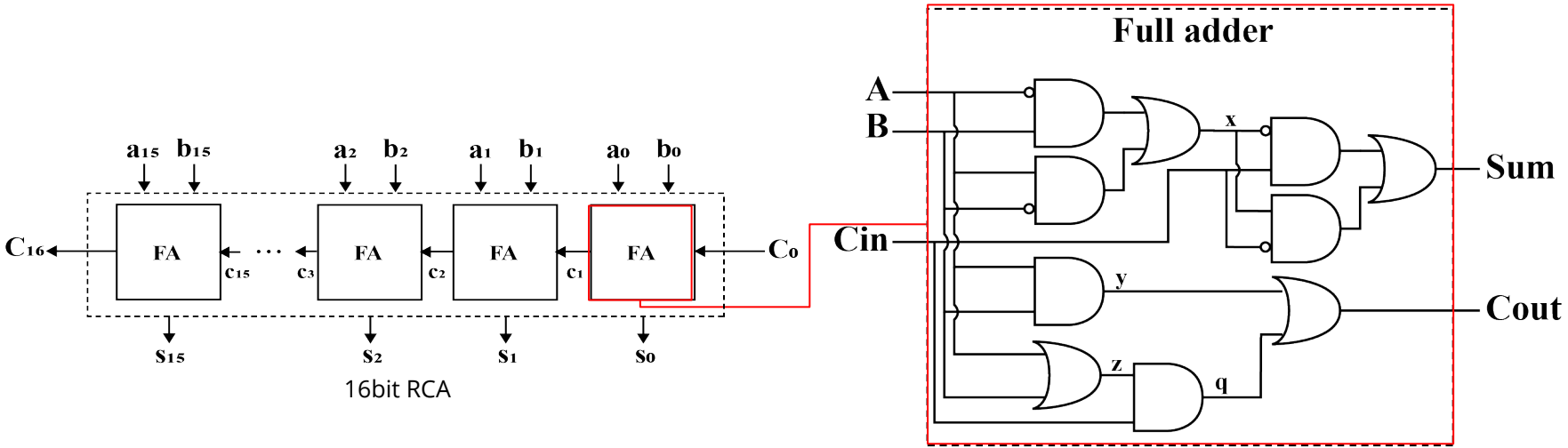
Procedural assignment

```
reg [15:0] sum;
reg cout;

always @(a or b or cin)  begin
      {cout,sum} = a + b + cin;
end
```

RCA w/ Verilog structural modeling

Ripple Carry Adder

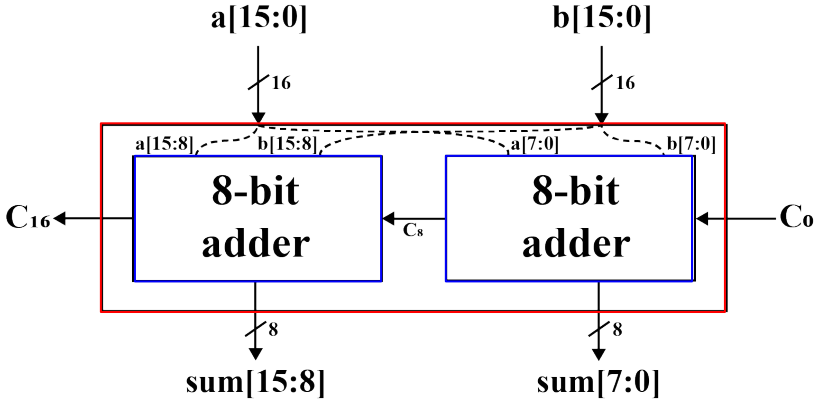
- Ripple carry adder (RCA) 是由多個加法器 (full adder) 連續組合而成, 其中第 n 個加法器必須等待前一個加法結果傳入後才能進行運算, 如同水波一樣依序傳遞 , 如下圖所示, 以下將以 16-bit RCA 介紹 structural modeling



Verilog structural modeling

- Structural modeling 是以邏輯閘層次描述的行為，透過 I/O 將 module 與互相連結

- 以右圖的行為區塊為例，16-bit RCA 可分為兩個 8-bit RCA，並且可將 8-bit RCA 再以兩個4-bit RCA、四個2-bit RCA 或八個加法器互相連結



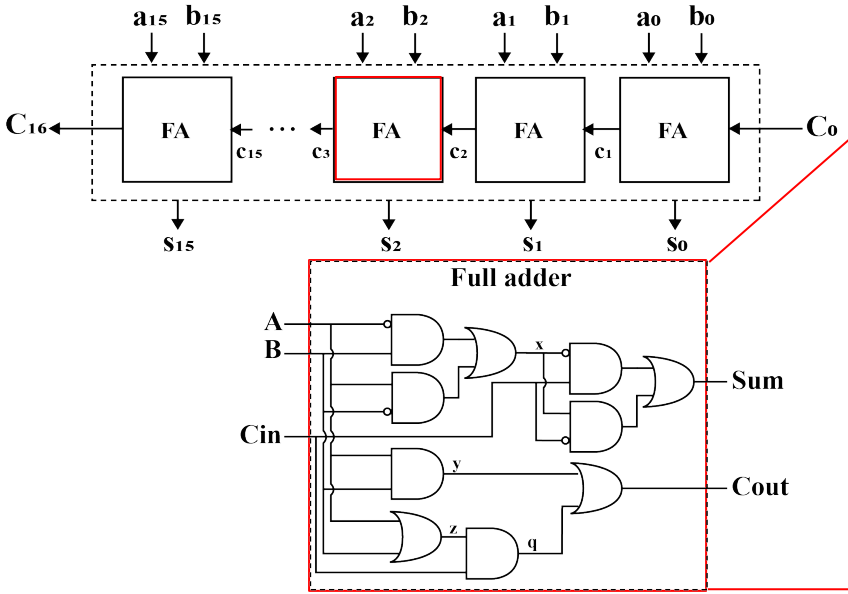

```
module RCA_16bit(a,b,sum,c0,c16);

        input [15:0] a,b;
        output [15:0] sum;
        input c0;
        output c16;
        wire carry;

        RCA_8bit FA1(a[7:0],b[7:0],sum[7:0],c0,carry);
        RCA_8bit FA2(a[15:8],b[15:8],sum[15:8],carry,c16);

endmodule
```

```
module RCA_8bit(a,b,sum,cin,cout);

        input [7:0] a,b;
        output [7:0] sum;
        output cin;
        output cout;
        wire [7:0] c;

        structural modeling design ...

        assign cout = c[7];
endmodule
```

Gate delay & Timing simulation

Gate delay & Timing simulation (1/4)

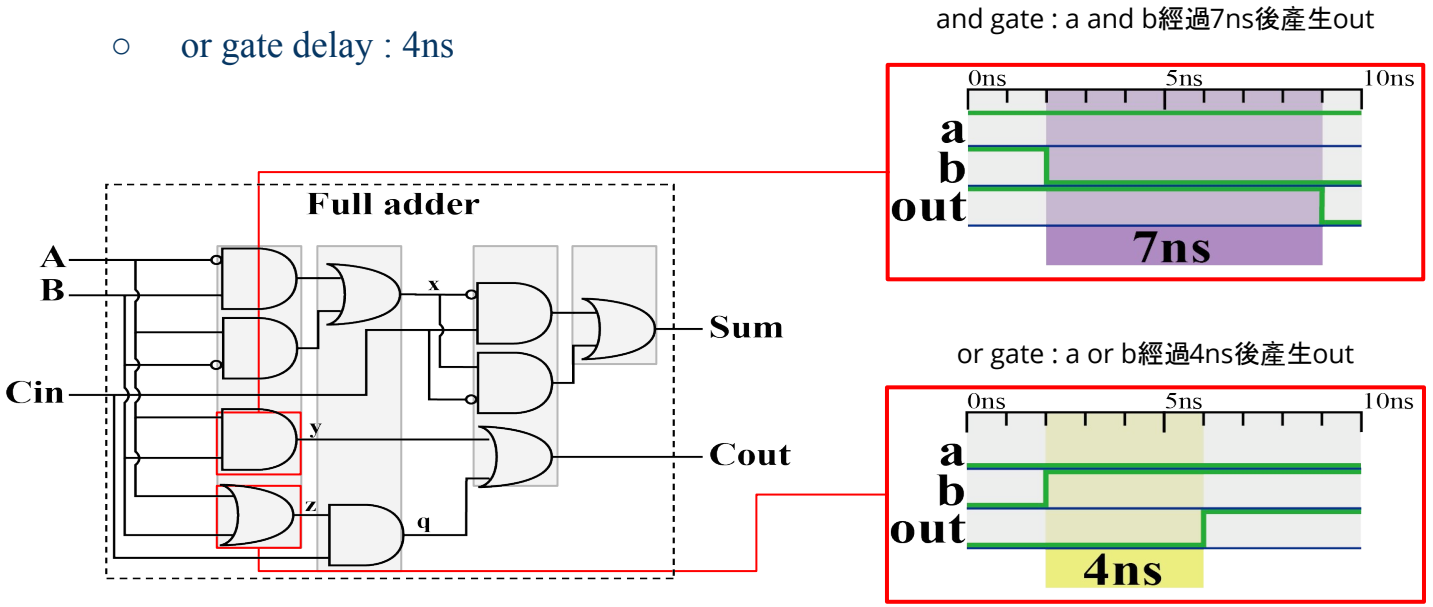
- 16-bit RCA 其中每個加法器的架構如圖下, 由各兩個 xor、and、or gate 組成, 圖右為其行為描述區塊




```
module fulladder(a,b,cin,sum,cout);

    input a, b, cin;
    output sum,cout;
    wire x,y,z,q;

    //sum = (a ^ b) ^ cin;
    //cout = ((a & b) | ((a | b) & cin));
    xorgate xor1 (a,b,x);
    xorgate xor2(x,cin,sum);
    andgate and1(a,b,y);
    andgate and2(z,cin,q);
    orgate or1(a,b,z);
    orgate or2(y,q,cout);

endmodule
```

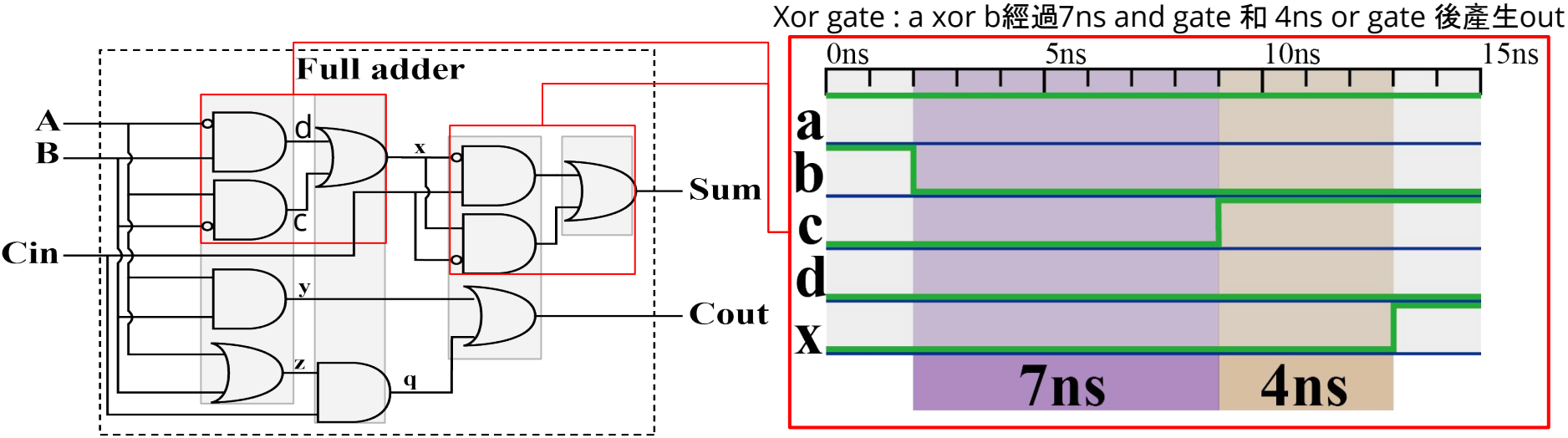
Gate delay & Timing simulation (2/4)

- 在電路中輸入的訊號每經過一個邏輯閘 (gate) 就會延遲一小段時間, 稱為 gate delay。觀察下圖加法器的波形, 以下將會對加法器 內的邏輯閘進行描述
  - and gate delay : 7ns
  - or gate delay : 4ns



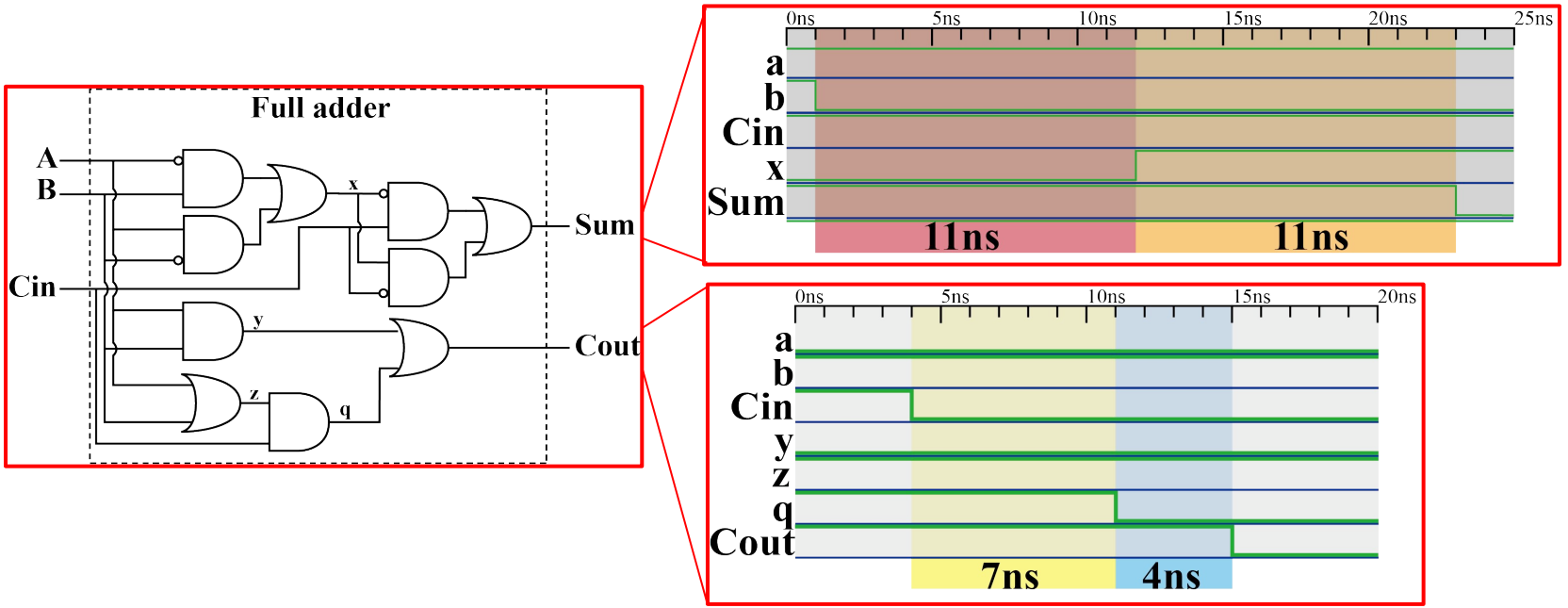
Gate delay & Timing simulation (3/4)

- 加法器其中有兩個 xor gate , 每個 xor gate 由兩個 and gate 和一個 or gate 組合而成, 其波形和行為描述區塊如下



Gate delay & Timing simulation (4/4)

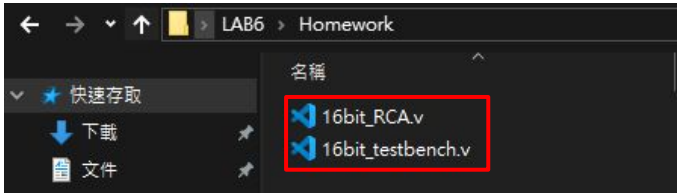
- 以下為 full adder 的完整 waveform, 在下圖可以觀察輸入值 A、B、Cin 到輸出值 Sum、Cout 的時序變化



實驗範例 & 作業

課堂範例

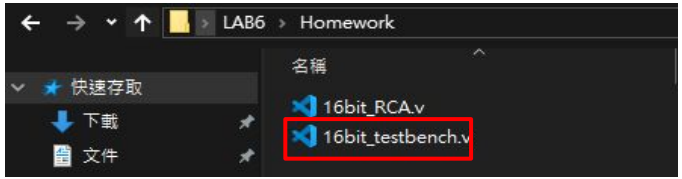
- 請參考投影片以及上課 內容, 開啟"Example"資料夾, 在 16bit_RCA.v 中透過 structural modeling 以範例提供的and gate 和 or gate 組合出 16-bit RCA, 使用 Verilog 驗證 testbench 的正確性, 若運算正確則會顯示如下圖結果, 總共有十道 test



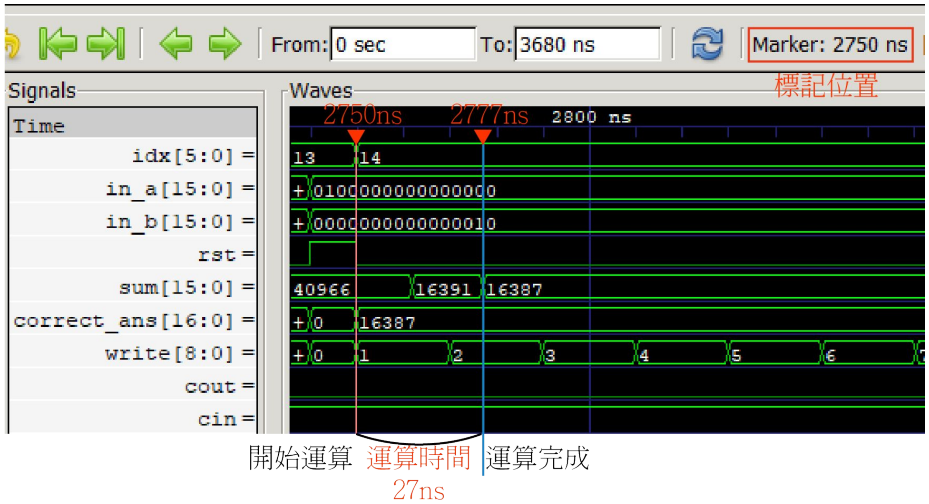

```
PS C:\Users\ECL\Desktop\Example> iverilog -o test .\16bit_RCA.v .\16bit_testbench.v
PS C:\Users\ECL\Desktop\Example> vvp .\test
VCD info: dumpfile lab06_16bit.fsdb opened for output.
////////////Test  1////////////
//  Q :18233 +  7947 + 1 = ? //
////////////////////////////////
//  Your answer               //
//  Cout = 0 Sum = 26181      //
////////////////////////////////
//  Correct answer            //
//  Cout = 0 Sum = 26181      //
////////////////////////////////
//        SUCCESSFUL !        //
////////////////////////////////
```

實驗作業

- 請將於範例完成的 16bit_RCA.v 移至"Homework"資料夾, 參考範例的 testbench , 修改 16bit_testbench.v 後使用 gtkwave 觀察波形, 在十道 test 內至少找出各一組輸入 (in_a,in_b,cin) 使 16-bit RCA 消耗最長 (worst case) 和最短 (best case) 運算時間, 最後在 gtkwave 中拉出下列訊號線並與助教說明你的答案



idx : 運算筆數
in_a、in_b、cin : 輸入值
sum ：你設計的RCA 運算結果
correct_ans : 正確答案



評分方式

- Demo時間 : 依E-Course公布為準
- Demo梯次 : 依E-Course公布為準
- Demo地點 : 501A
- 課程評分方式
  - 範例 : 展示你的 structural modeling 設計以及 testbench 的十道運算結果 (60%)
  - 作業 : 至少各找出一組 worst case 和 best case 並展示 gtkwave 波形給助教查看 (40%)
- 請記得填寫意見回饋表, 否則不予計分

- 若對本次範例、作業或評分方式有任何疑問, 請寄信到本次 lab 的助教信箱詢問, 謝謝
  - 徐孟澤 sszrop321@gmail.com
  - 簡睿宇 ru03bjo4m385122@gmail.com