

DD LAB9 : Sequential MPY

助教：東昇、憲億、琮閔、韋廷、文駿

Outline

- 課程目的
- Sequential Circuit 介紹
- 課堂練習— Serial Multiplier
- Lab作業說明
- 作業繳交 & demo事項

課程目的

- 先前的實驗課程已經教導各位如何利用structural modeling的技巧來實現硬體架構，本次實驗要教各位利用behavioral modeling 的技巧來撰寫 sequential circuit

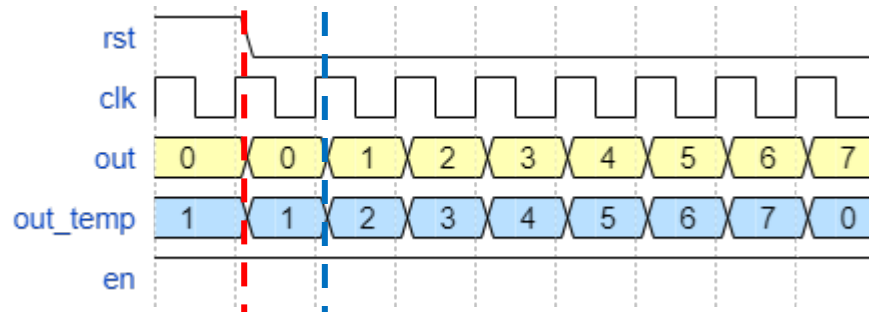
Sequential Circuit 介紹

```
module Counter(clk, rst, en, out);
    input clk, rst, en;
    output reg [2:0] out;
    reg [2:0] out_temp;

    //sequential circuit
    always @(posedge clk) begin
        if(rst)
            out <= 3'b0;
        else begin
            if(en)
                out <= out_temp;
            else
                out <= out;
        end
    end

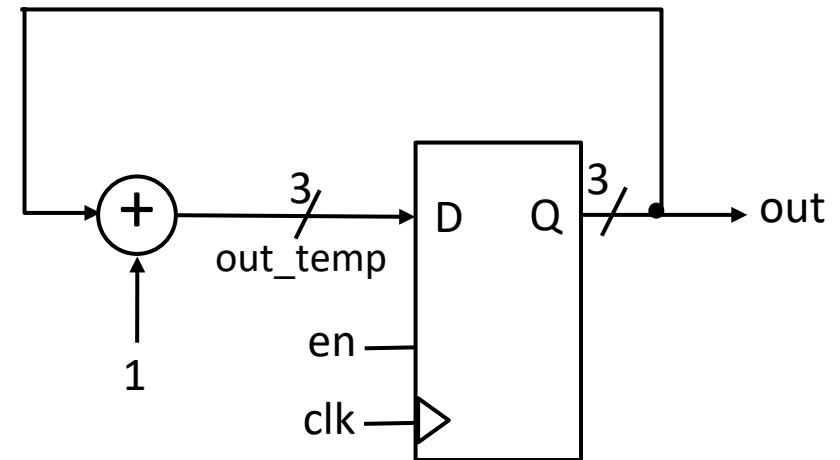
    //combinational circuit
    always @(*)begin
        out_temp = out + 1'b1;
    end
endmodule
```

• 範例程式



• 波形圖

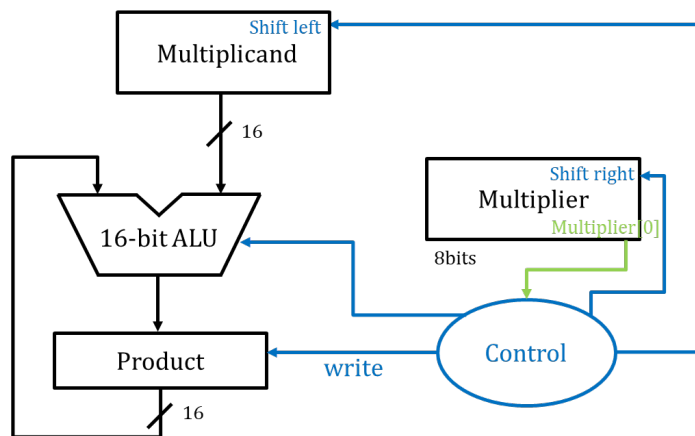
- 這邊以一個3-bit Counter當作範例說明
- rst 初始為1 → 執行out <= 3'b0
- rst = 0, en = 1 → out_temp存入out並在下次posedge clk輸出



• 範例程式架構圖

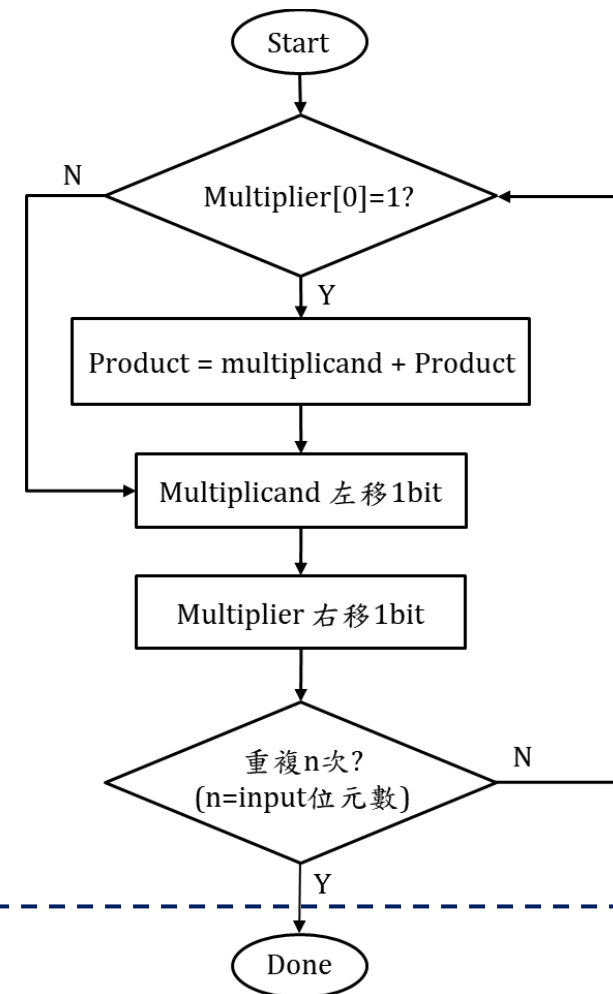
課堂練習 - Serial Multiplier

- 助教會提供此架構的程式“lab9.v”，在下一頁會進行說明
- 下方提供Serial Multiplier的流程圖與架構示意圖



- 這邊以 4-bit input 當範例呈現運算過程

n	Product	multiplier	multiplicand
0	0000_0000	0011	0000_0010
1	0000_0010	0001	0000_0100
2	0000_0110	0000	0000_1000
3	0000_0110	0000	0001_0000
4	0000_0110	0000	0010_0000
<hr/>			
	0000_0110		



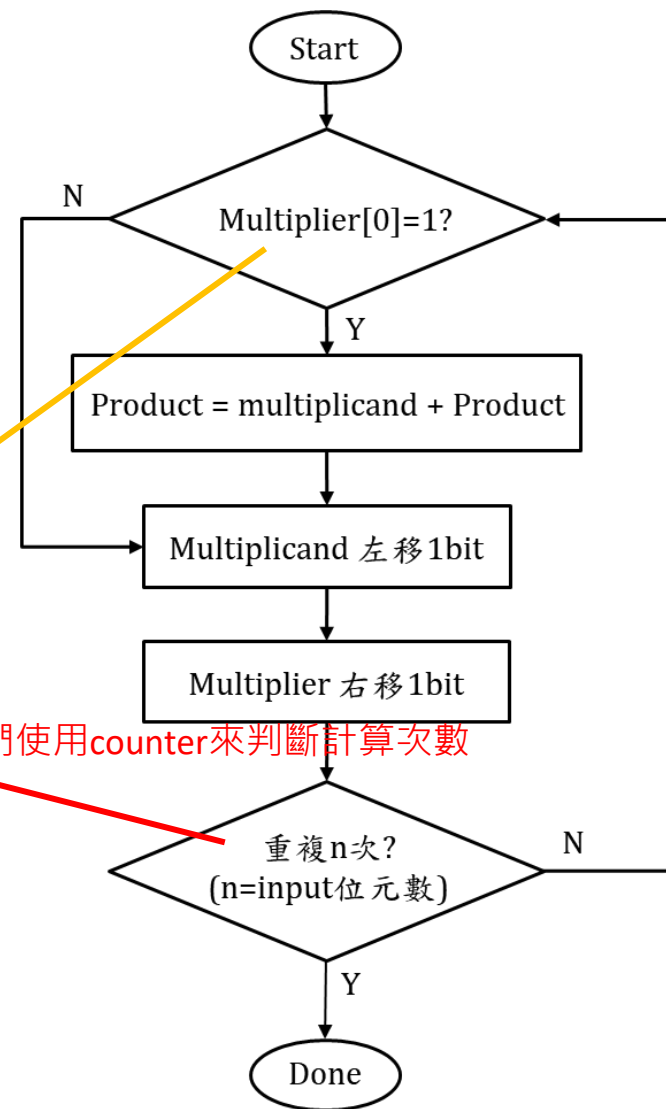
Serial Multiplier Implement

```
always @(posedge CLK or posedge RST)
begin
    if(RST) begin
        Product <= 16'b0;
        Mplicand <= 16'b0;
        Mplier <= 8'b0;
    end

    else if(Counter == 6'd0) begin 初始化
        Product <= 16'b0;
        Mplicand <= {16'b0,in_a};
        Mplier <= in_b;
    end

    else if(Counter <= 6'd8)
    begin
        if(Mplier[0] == 1'b1)
        Product <= Mplicand + Product;
        Mplicand <= Mplicand << 1'b1;
        Mplier <= Mplier >> 1'b1;
    end 每次計算都需位移

    else begin
        Product <= Product;
        Mplicand <= Mplicand;
        Mplier <= Mplier;
    end
end
```



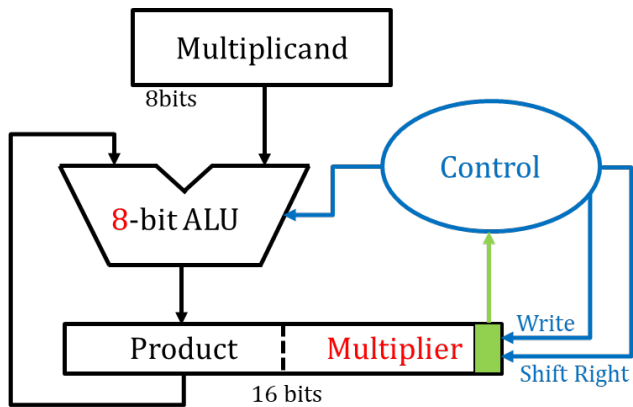
我們使用counter來判斷計算次數

Lab作業說明

- 透過上述的練習我們已經學會簡單的Sequential Circuit，接下來請大家參考範例，實作將前述Serial Multiplier優化之Optimized Serial Multiplier與Serial Radix-4 Booth Multiplier

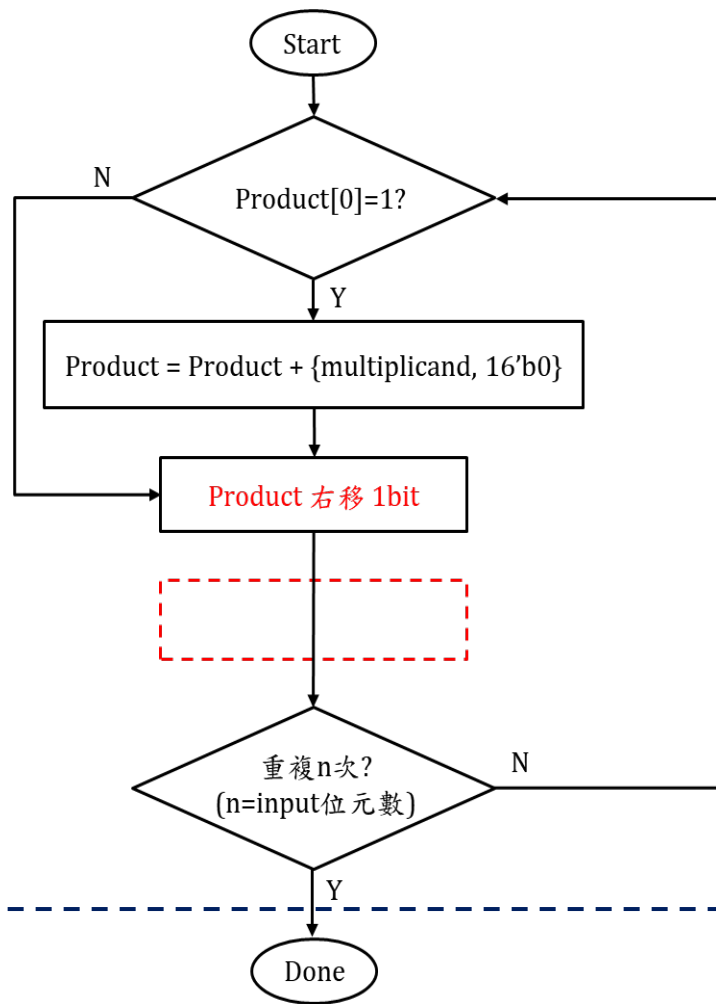
Optimized Serial Multiplier

- 我們縮減被乘數暫存器，並取消左移功能
- 乘積暫存器增加了右移功能並與乘數暫存器合併 {乘積, 乘數}



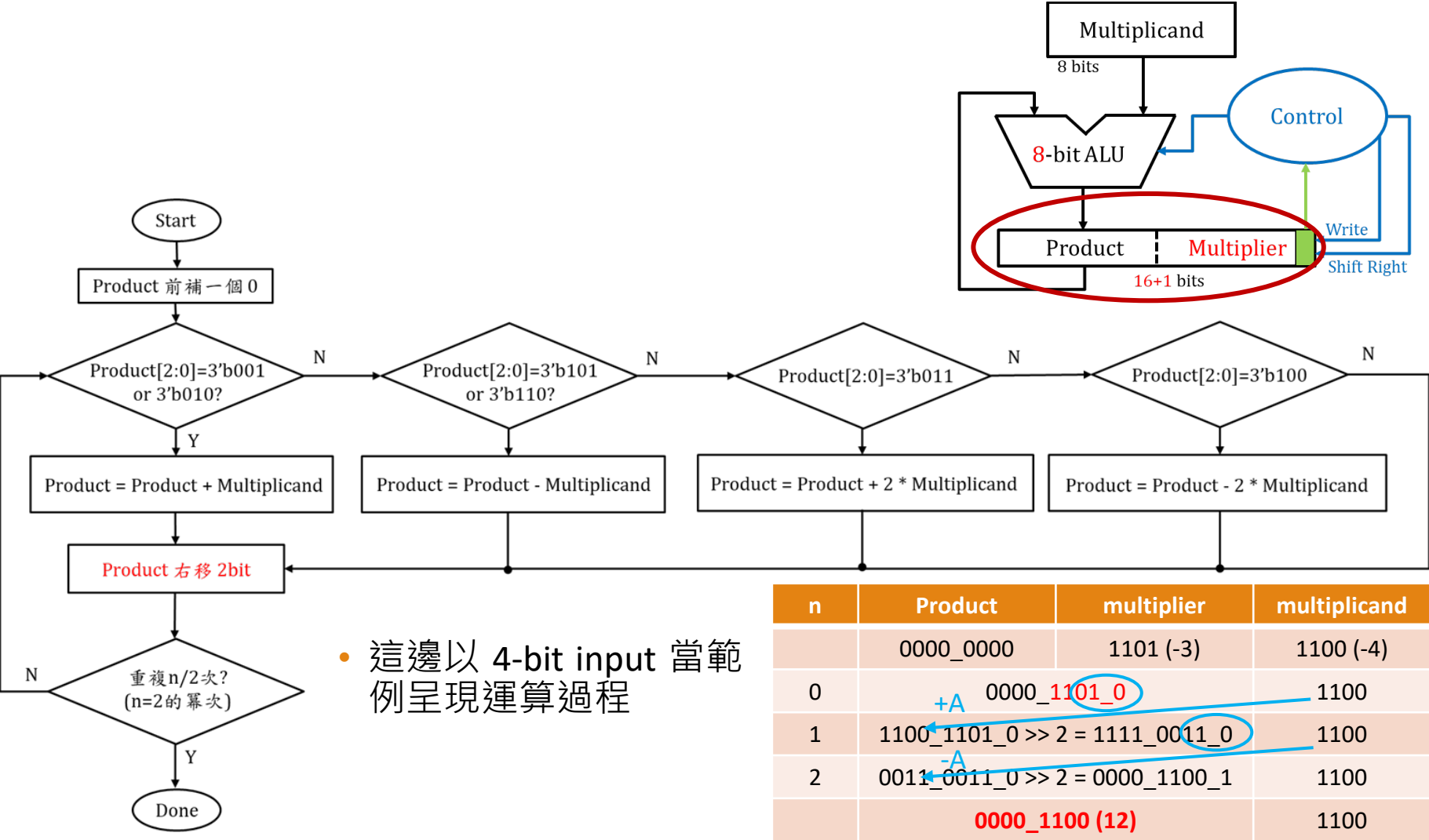
- 這邊以 4-bit input 當範例呈現運算過程

n	Product	multiplier	multiplicand
	0000_0000	0011	0010
0	0000_0011		0010
1	0010_0011 >> 1 = 0001_0001		0010
2	0011_0001 >> 1 = 0001_1000		0010
3	0001_1000 >> 1 = 0000_1100		0010
4	0000_1100 >> 1 = 0000_0110		0010
	0000_0110		0010



Serial Radix-4 Booth Multiplier

- 使用上一頁的架構，並以sequential circuit的方式實現Lab8的Radix-4 Booth Multiplier，以下附上架構圖、流程圖以及範例



這邊以 4-bit input 當範例呈現運算過程

LAB 作業

- 在不更動 testbench 的前提下，修改範例程式“lab9.v”為 Optimized Serial Multiplier 及 Radix-4 Booth Multiplier
- 成功執行 tb_lab9_hw_unsigned.v

作業繳交 & demo事項

- 作業繳交時間
 - 2021/6/16(三) 23:59前，上傳至eCourse2
- 作業上傳檔案
 - 1.Optimized Serial Multiplier 的 Design
 - 2.Radix-4 Booth Multiplier 的 Design
 - 3.testbench(tb_lab9_hw_unsigned.v)
- Demo時間
 - 於社團日後公告
- Demo地點
 - 資工館501A實驗室
- 可在Demo時間前5分鐘至Demo地點準備
- 請攜帶作業相關檔案(使用隨身碟，不要使用雲端)
- 可使用自己的筆電demo