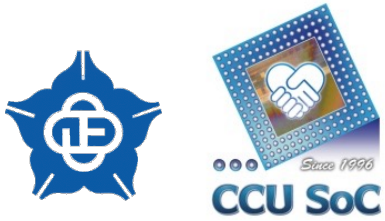
DD LAB9 :
Sequential Circuit & Serial Multiplier

助教：林冠翰、徐瑋程

Department of Electrical Engineering and SoC Research Center
National Chung Cheng University



Outline

- 課程目的
- Sequential Circuit 介紹
- 課堂練習— Serial Multiplier
- Lab作業說明
- 課程評分

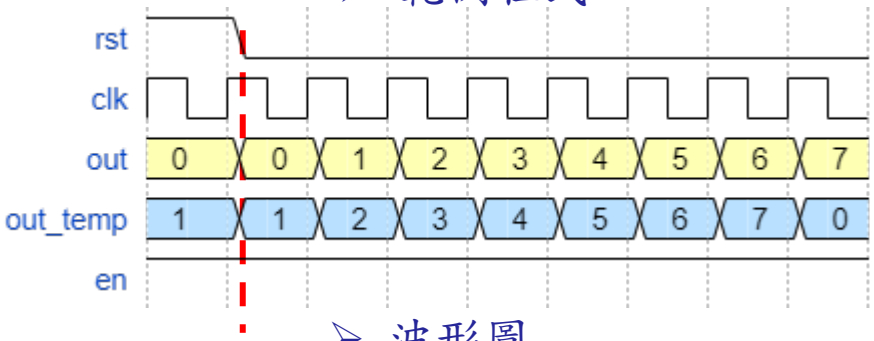
課程目的

- 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

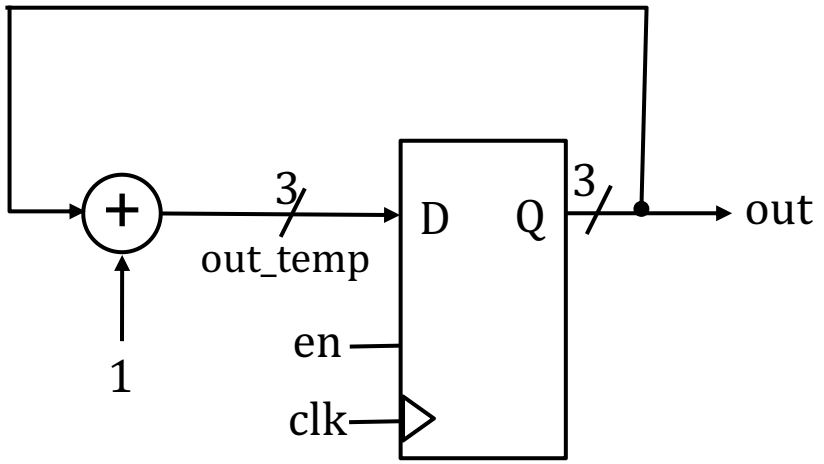
```
module Counter(clk, rst, en, out);
    input clk, rst, en;
    output reg [2:0] out;
    reg [2:0] out_temp;

    //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

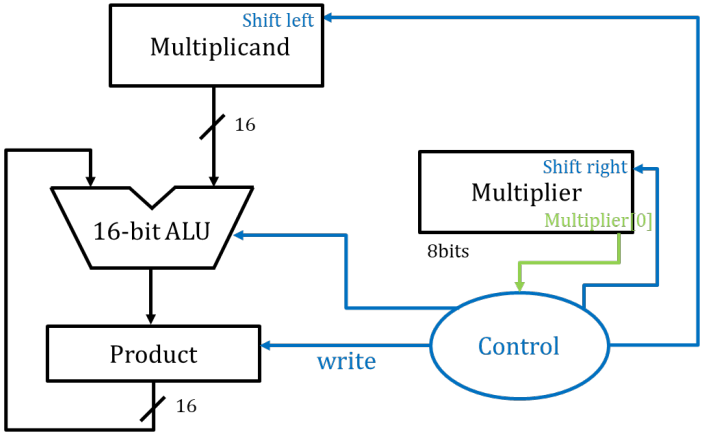
- ❖ 這邊以一個3-bit Counter當作範例說明
- ❖ rst初始為0 → 執行out <= 3'b0
- ❖ rst = 1 → out_temp存入out並在下一次posedge clk輸出

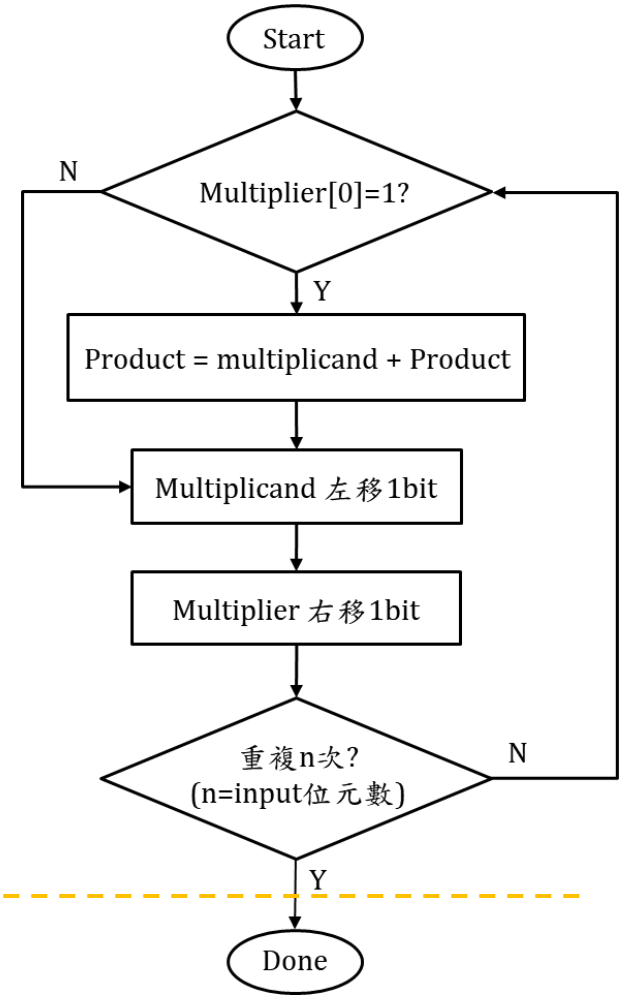


➢ 範例程式架構圖

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

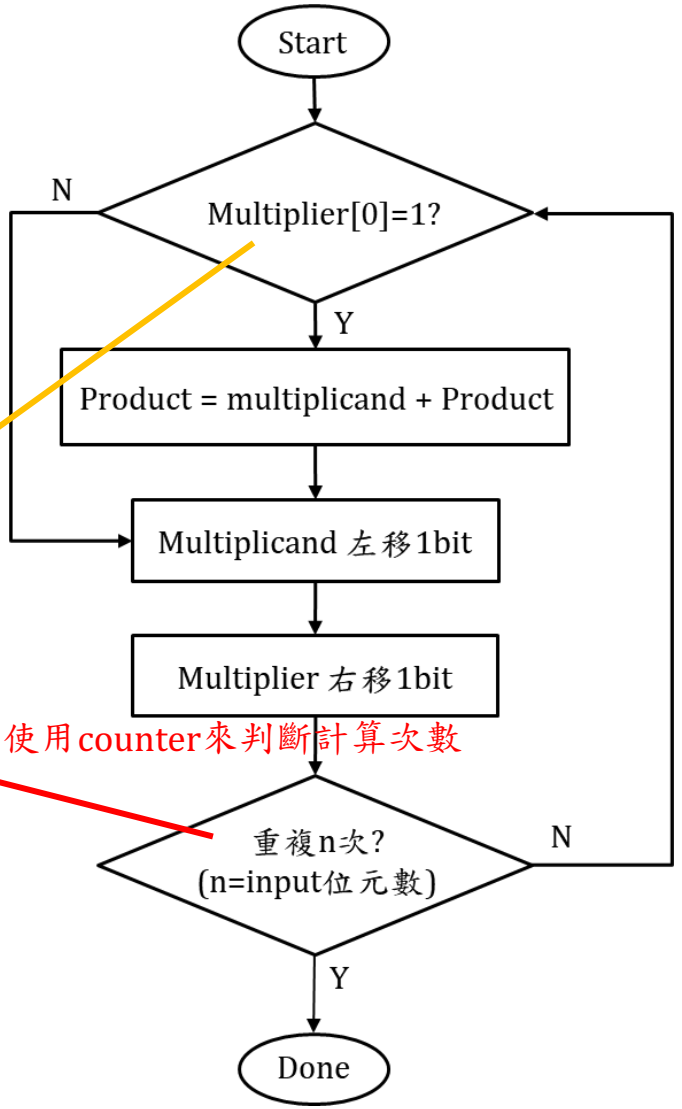
n	Product	multiplier	multiplicand
0	0000_0000	0011 +	0000_0010
1	0000_0010	0001 +	0000_0100
2	0000_0110	0000	0000_1000
3	0000_0110	0000	0001_0000
4	0000_0110	0000	0010_0000
	0000_0110		

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

初始化

每次計算都需位移



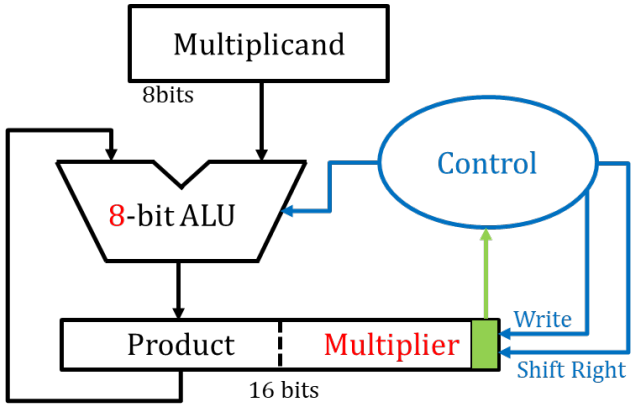
我們使用counter來判斷計算次數

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，並在FPGA上執行

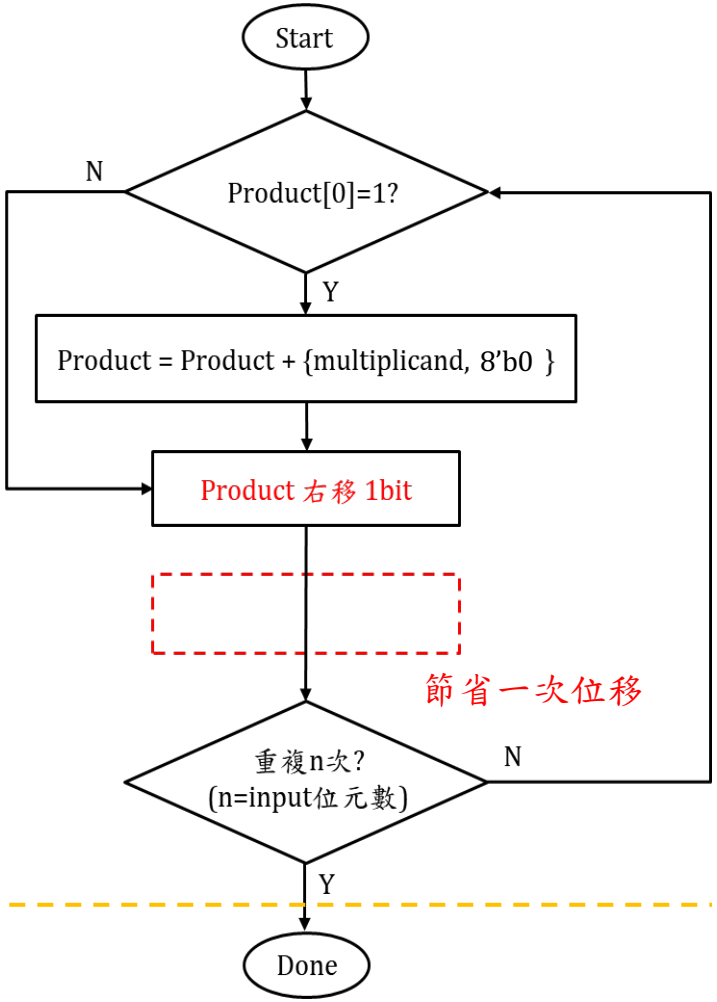
Optimized Serial Multiplier

- 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}



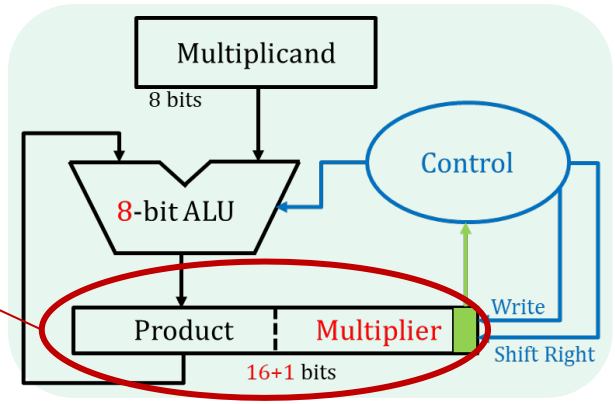
- 這邊以 4-bit input 當範例呈現運算過程

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

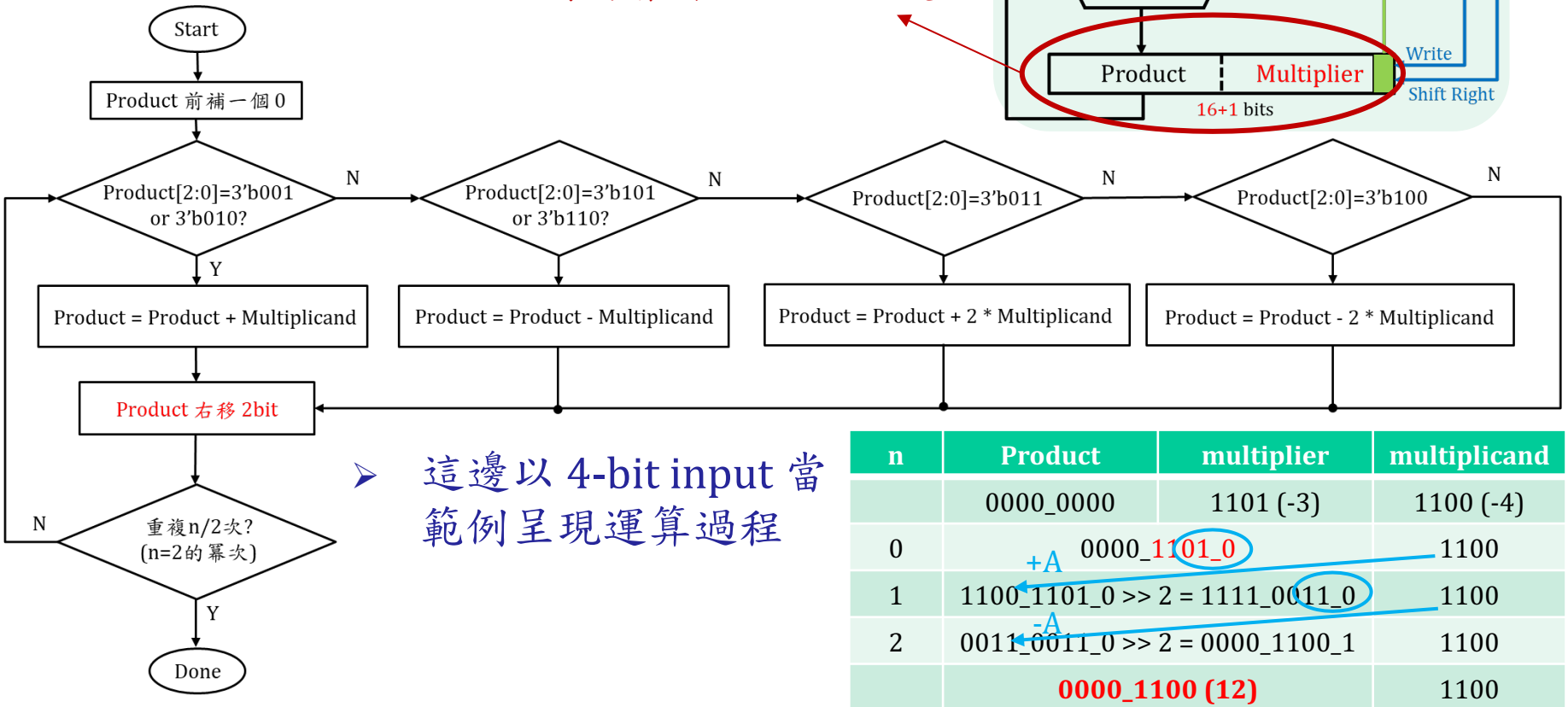


Serial Radix-4 Booth Multiplier

- 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



demo時要看到此16bits的變化



- 這邊以 4-bit input 當範例呈現運算過程

n	Product	multiplier	multiplicand
	0000_0000	1101 (-3)	1100 (-4)
0	0000_1101_0	+A	1100
1	1100_1101_0 >> 2 = 1111_0011_0		1100
2	0011_0011_0 >> 2 = 0000_1100_1	-A	1100
	0000_1100 (12)		1100

回家作業與配分

1. 在不更動 testbench的前提之下，修改範例程式"lab9.v" 為Optimized Serial Multiplier (60%)
   - 成功執行 tb_lab9_hw_unsigned.v
2. 實作8 x 8 Serial Radix-4 Booth有號數乘法器並使用七段顯示器顯示結果 (40%)
   - switch[7:0], switch[15:8] 分別為兩個有號數input
   - 兩數相乘的結果在七段顯示器上顯示
   - Button (M18)當作reset
   - Button (N17)用來開始乘法運算
   - demo時要看到每個cycle值的變化(總共16bits不包含用來幫助判斷booth的1bit)

記得填寫意見回饋表，否則不予以計分

附錄：Radix-4 Booth

❖ Radix-4 booth 的規則如下表

b_{i+1}	b_i	b_{i-1}	operation
0	0	0	0
0	0	1	+A
0	1	0	+A
0	1	1	+2A
1	0	0	-2A
1	0	1	-A
1	1	0	-A
1	1	1	0