

# 數位系統設計實習

## Lecture 6

指導老師：陳勇志 教授  
實習課助教：蔡沛希

# Outline

2

- Shift Register
- 4-Bit Universal Shift Register
- Switch-Tail Ring Counter
- Johnson Counter
- LAB 6-1 ~ 6-3

# Chapter

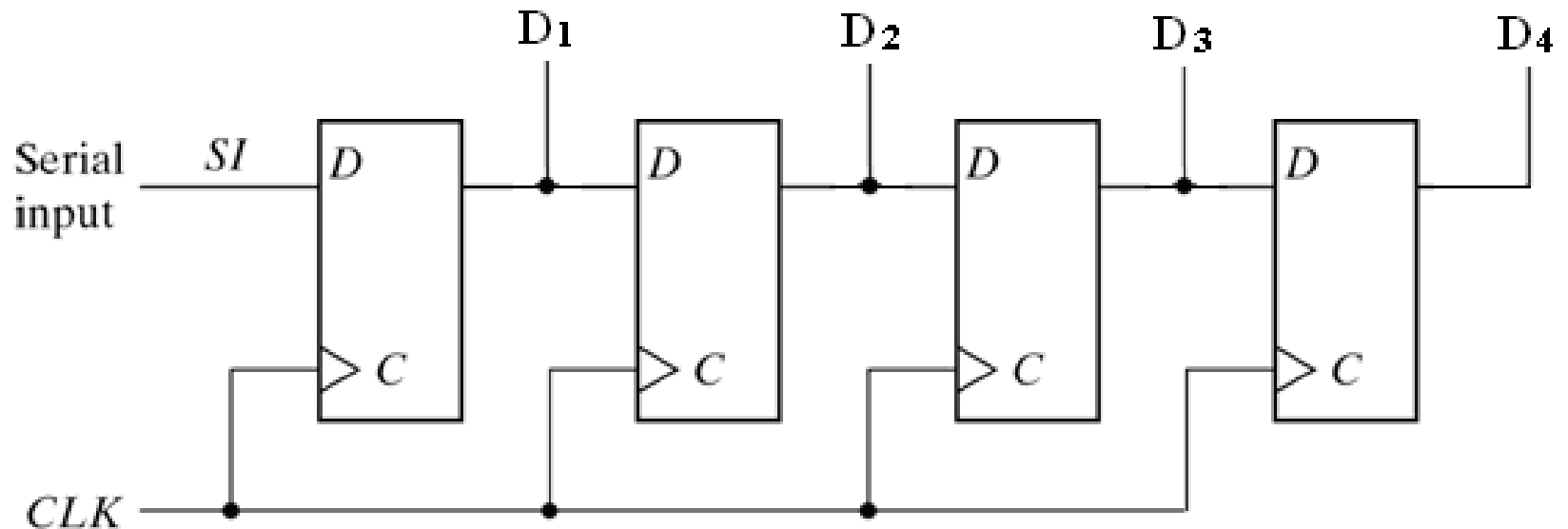
3

- Shift Register
- 4-Bit Universal Shift Register
- Switch-Tail Ring Counter
- Johnson Counter
- LAB 6-1 ~ 6-3

# Shift Register

- Shift Register 的邏輯結構是由一串連接而成的正反器所組成，一個正反器的輸出接到下個正反器的輸入，所有的正反器接受共同的時脈，促成資料由某一級移位至下一級。
- 4-bit Serial-In Parallel-Out Shift Register，當 CLK 正源觸發，Serial input 進入最左邊暫存器，D1、D2、D3、D4 平行輸出移位暫存器內的值。

# Shift Register



4-bit Serial In Parallel Out Shift Register

# Chapter

6

- Shift Register
- 4-Bit Universal Shift Register
- Switch-Tail Ring Counter
- Johnson Counter
- LAB 6-1 ~ 6-3

# 4-Bit Universal Shift Register

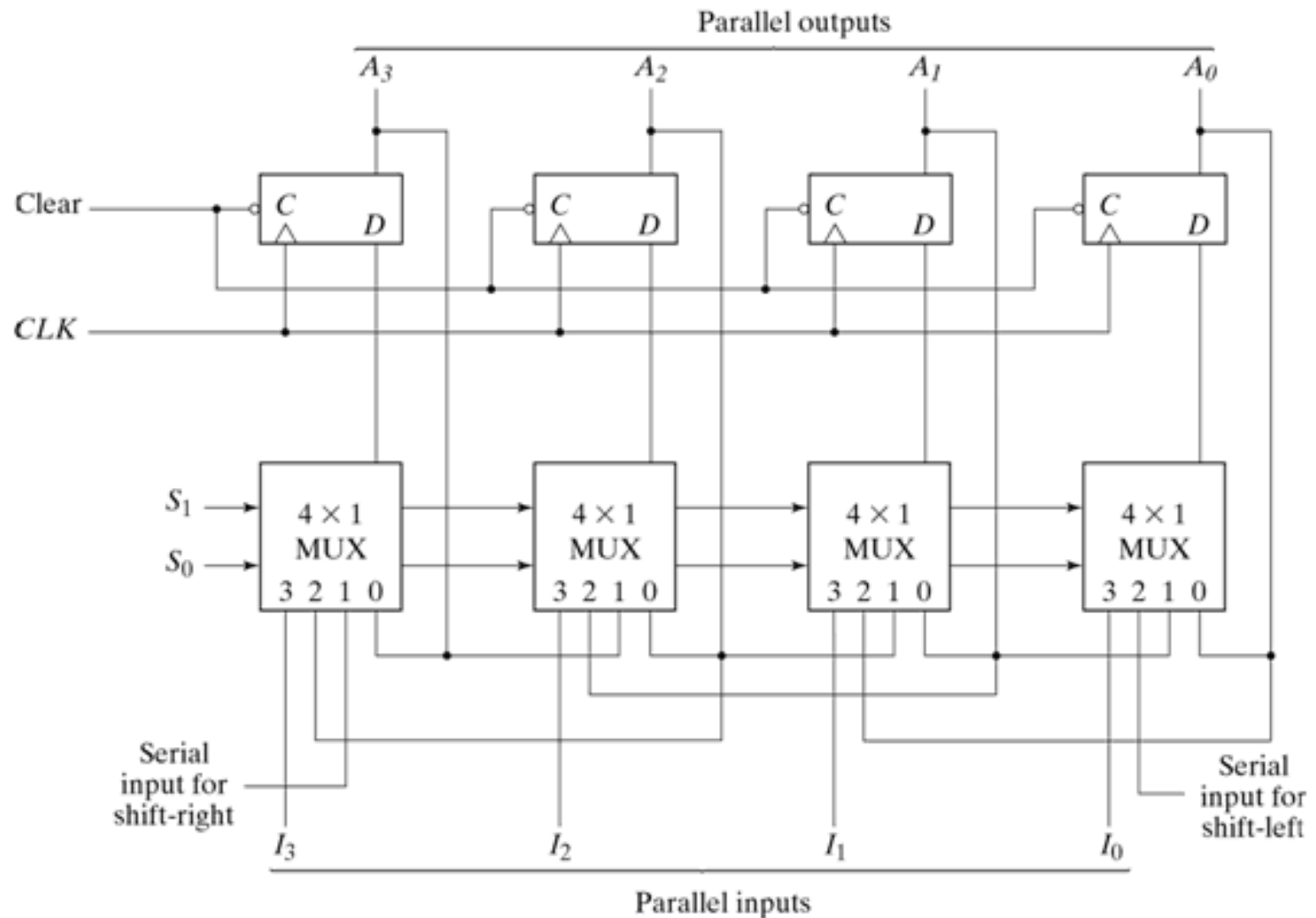
- 4-Bit Universal Shift Register，輸入 CLK，使所有操作同步化；Clear 為清除控制，將暫存器清除為 0；兩個 serial inputs 分別為 shift-right 及 shift-left。
- $I_0$ 、 $I_1$ 、 $I_2$  和  $I_3$  為 parallel inputs； $S_1$  及  $S_2$  為輸入用來控制 Shift Register 的狀態為右移、左移、平行載入或者是保持原來的狀態。

# 4-Bit Universal Shift Register

模式控制		記錄器 操作
$s_1$	$s_2$	
0	0	不變
0	1	右移
1	0	左移
1	1	並列載入



# 4-Bit Universal Shift Register



# Shift Register

```
module shftreg (s1,s0,Pin,lfin,rtin,A,CLK,Clr);
input s1,s0;           //Select inputs
input lfin, rtin;      //Serial inputs
input CLK,Clr;         //Clock and Clear
input [3:0] Pin;       //Parallel input
output [3:0] A;        //Register output
reg [3:0] A;
always @ (posedge CLK or negedge Clr)
if (!Clr) A <= 4'b0000;
else
    case ({s1,s0})
        2'b00: A <= A;           //No change
        2'b01: A <= {rtin,A[3:1]}; //Shift right
        2'b10: A <= {A[2:0],lfin}; //Shift left
        2'b11: A <= Pin;         //Parallel load input
    endcase
endmodule
```

# Chapter

11

- Shift Register
- 4-Bit Universal Shift Register
- Switch-Tail Ring Counter
- Johnson Counter
- LAB 6-1 ~ 6-3

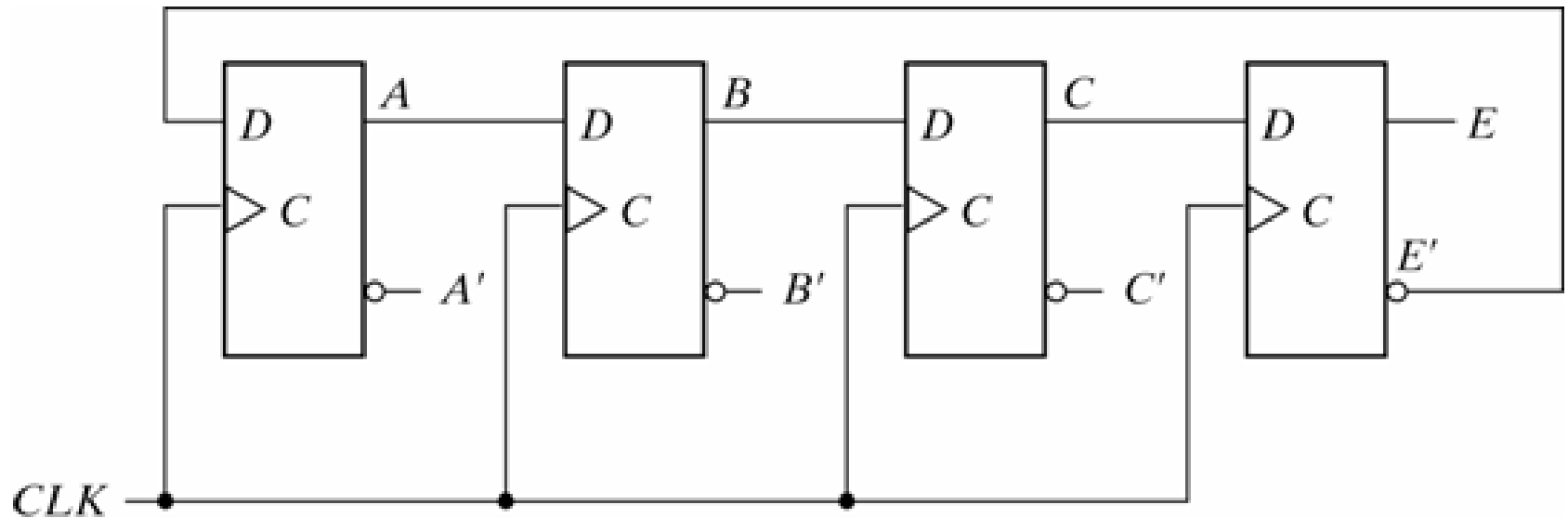
# K-Bit Ring-Counter

- 一個 K-Bit Ring-Counter，將每一個正反器的輸出連接到下一個正反器的輸入，最後一個正反器的輸出返回到第一個正反器，可以提供 K 個不同的輸出訊號態。

# Switch-Tail Ring Counter

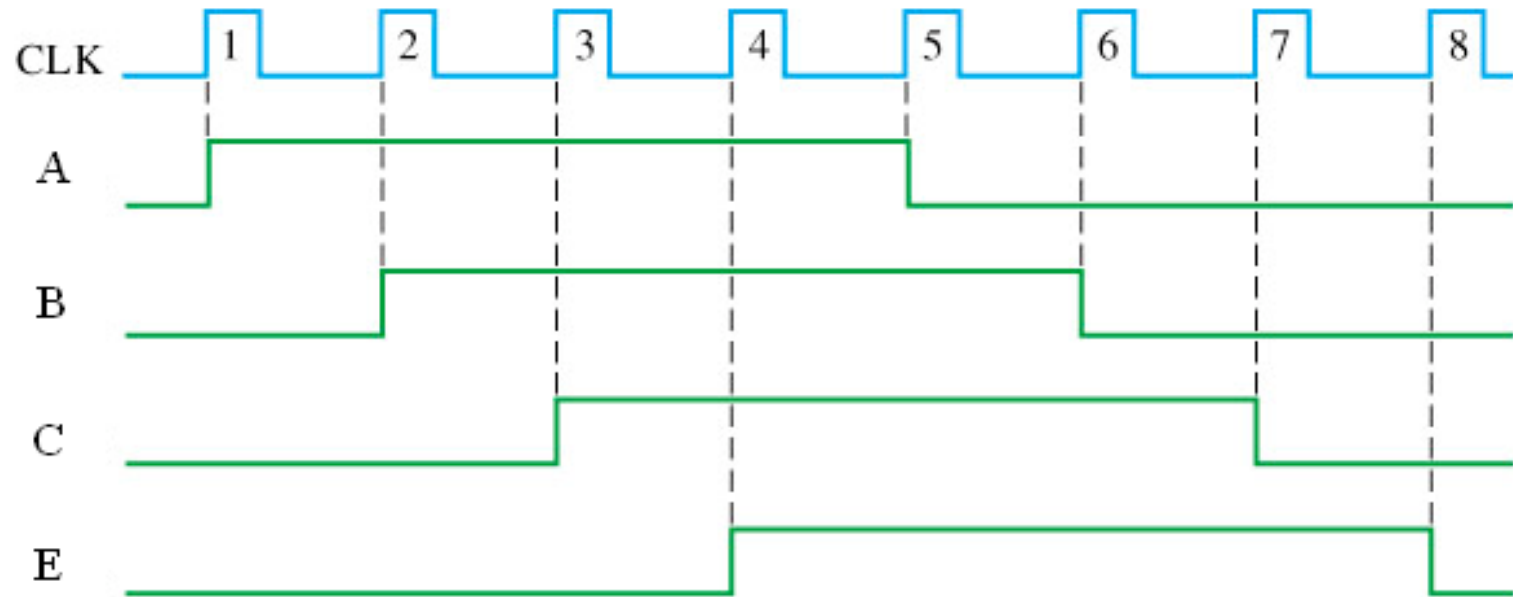
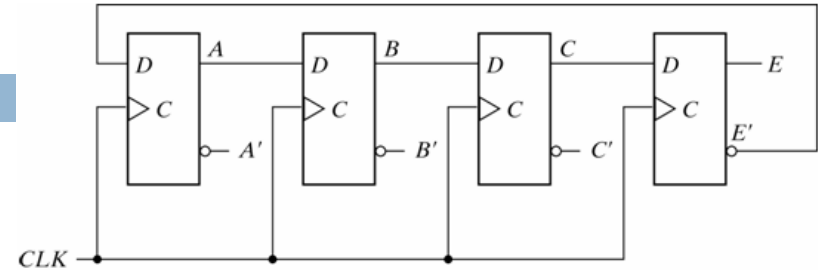
- 一個 K-Bit Ring-Counter 可以提供 K 個不同的輸出訊號狀態，但若將最後一個 stage 正反器輸出的反向連接到第一個 stage 的正反器的輸入，就是一個 Switch-Tail Ring Counter。
- 一個 K-Bit Switch-Tail Ring Counter 可以提供  $2K$  個不同的輸出訊號狀態。

# Switch-Tail Ring Counter



Four-Stage Switch-Tail Ring Counter

# Switch-Tail Ring Counter



Four-Stage Switch-Tail Ring Counter 之輸出波型圖

# Chapter

16

- Shift Register
- 4-Bit Universal Shift Register
- Switch-Tail Ring Counter
- Johnson Counter
- LAB 6-1 ~ 6-3



# Johnson Counter

- Johnson Counter 是由 Switch-Tail Ring Counter 及  $2K$  個 decoding gate 所組成。
- 圖7-1所列的 8 個 AND gate 加入 Switch-Tail Ring Counter 後就是一個完整的 Four-Stage Johnson Counter，其 8 個輸出波型如圖7-2所示。

# Johnson Counter

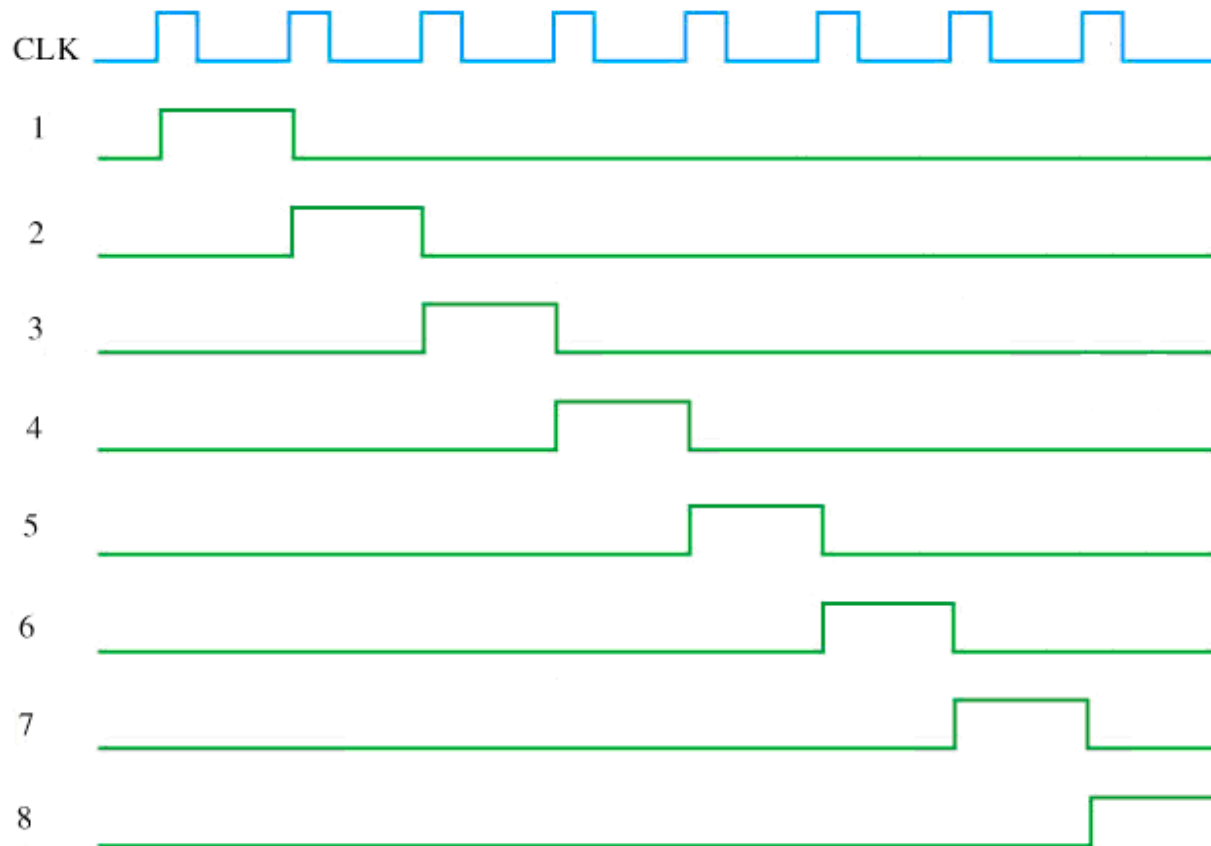
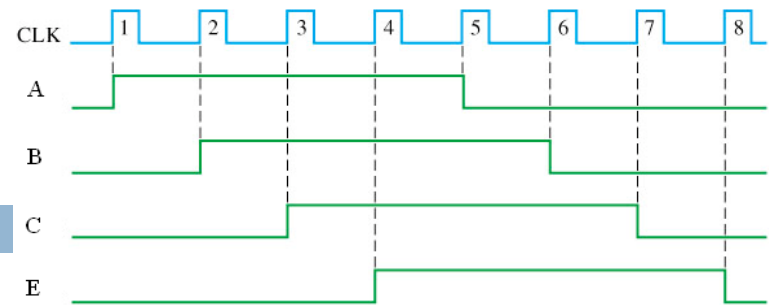


圖7-2

# Johnson Counter



Sequence number	Flip-flop outputs				AND gate required for output
	<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>	
1	1	0	0	0	$AB'$
2	1	1	0	0	$BC'$
3	1	1	1	0	$CE'$
4	1	1	1	1	$AE$
5	0	1	1	1	$A'B$
6	0	0	1	1	$B'C$
7	0	0	0	1	$C'E$
8	0	0	0	0	$A'E'$

圖 7-1

# Chapter

20

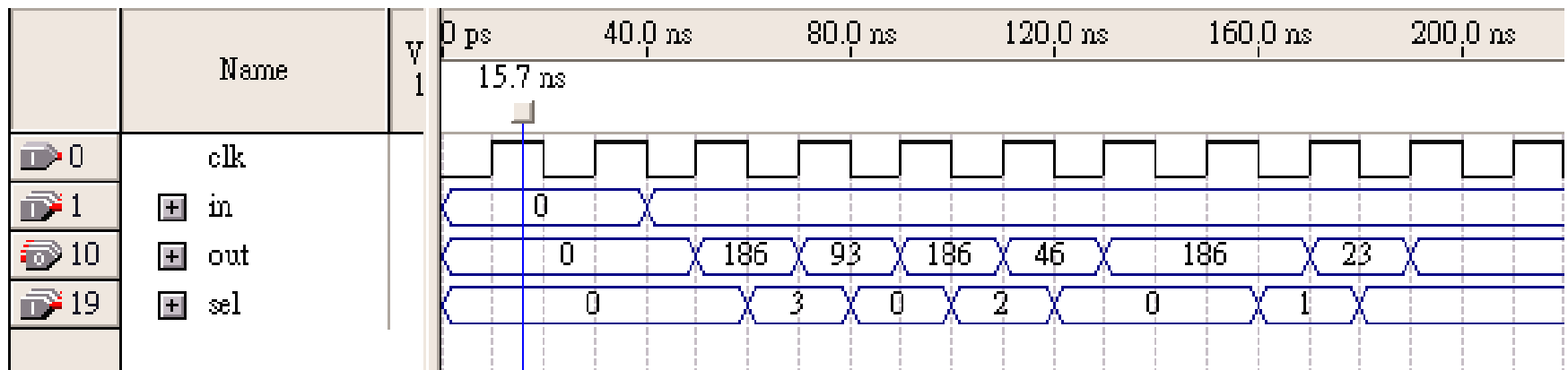
- Shift Register
- 4-Bit Universal Shift Register
- Switch-Tail Ring Counter
- Johnson Counter
- LAB 6-1 ~ 6-3

# LAB 6-1

- 請設計一個8位元移位暫存器，規格如下：
  - 當控制線 S1，S2 輸入為 00 時，平行載入；
  - 當控制線 S1，S2 輸入為 01 時，在一時脈內向右shift 3 位元；
  - 當控制線 S1，S2 輸入為 10 時，在一時脈內向右shift 2 位元；
  - 當控制線 S1，S2 輸入為 11 時，在一時脈內向右shift 1 位元。
  - PS：向右移時補0。

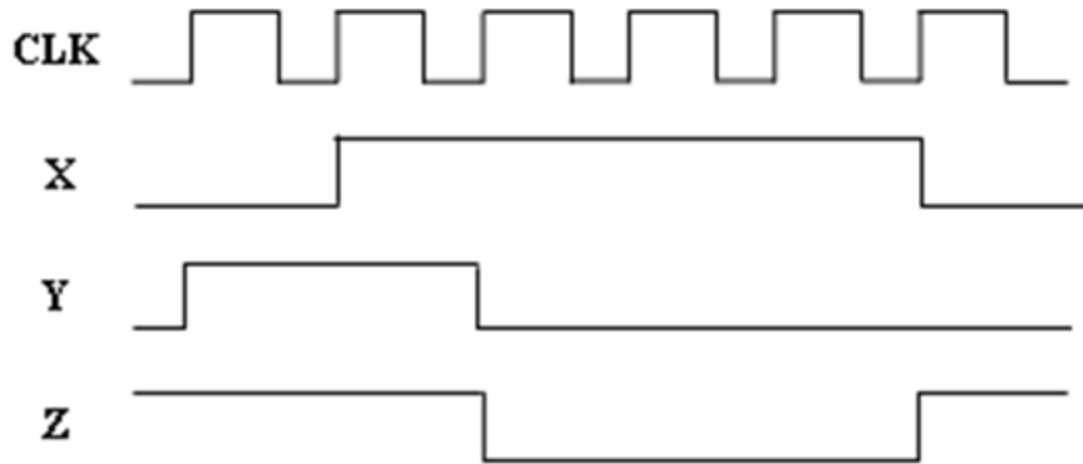
# LAB 6-1

22



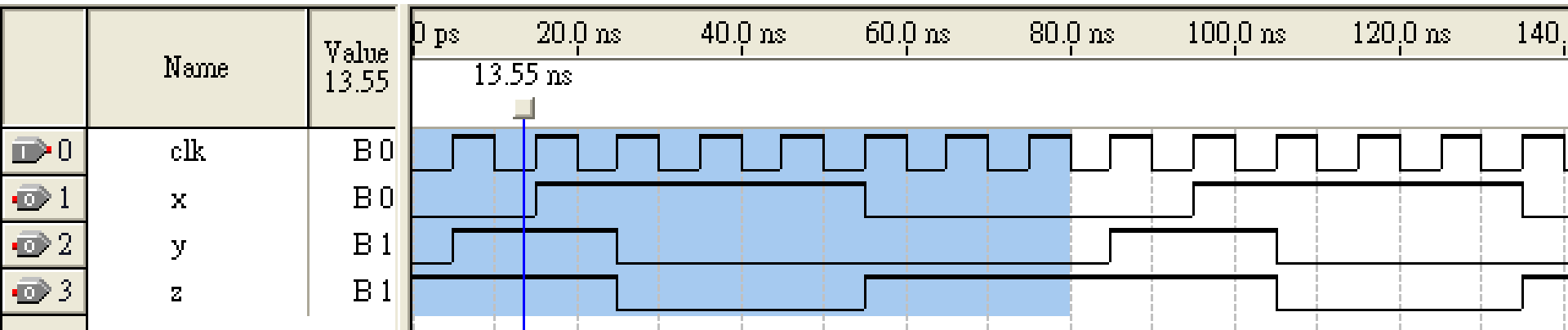
## LAB 6-2

- 請利用 Switch-Tail Ring Counter，於 Quartus II 上模擬出下列波型：



# LAB 6-2

24





# LAB 6-3

- 圖10.8為一 Serial Adder，請用 Verilog HDL 描寫，並於 Quartus II 模擬訊號波型加以驗證結果。

## HINT

- 1.Shift control 訊號線，控制 shift register 的動作，為 1 時資料向右 shift 1 位元，0 不動作。
- 2.當資料向右 shift 時，當 Serial input 為 1，則最左邊的 bit 補 1，反之，補 0。
- 3.全加法器的動作是去對 shift register A(p\_out1) 與 shift register B(p\_out2) 的第 0 個 bit 做相加。

# LAB 6-3

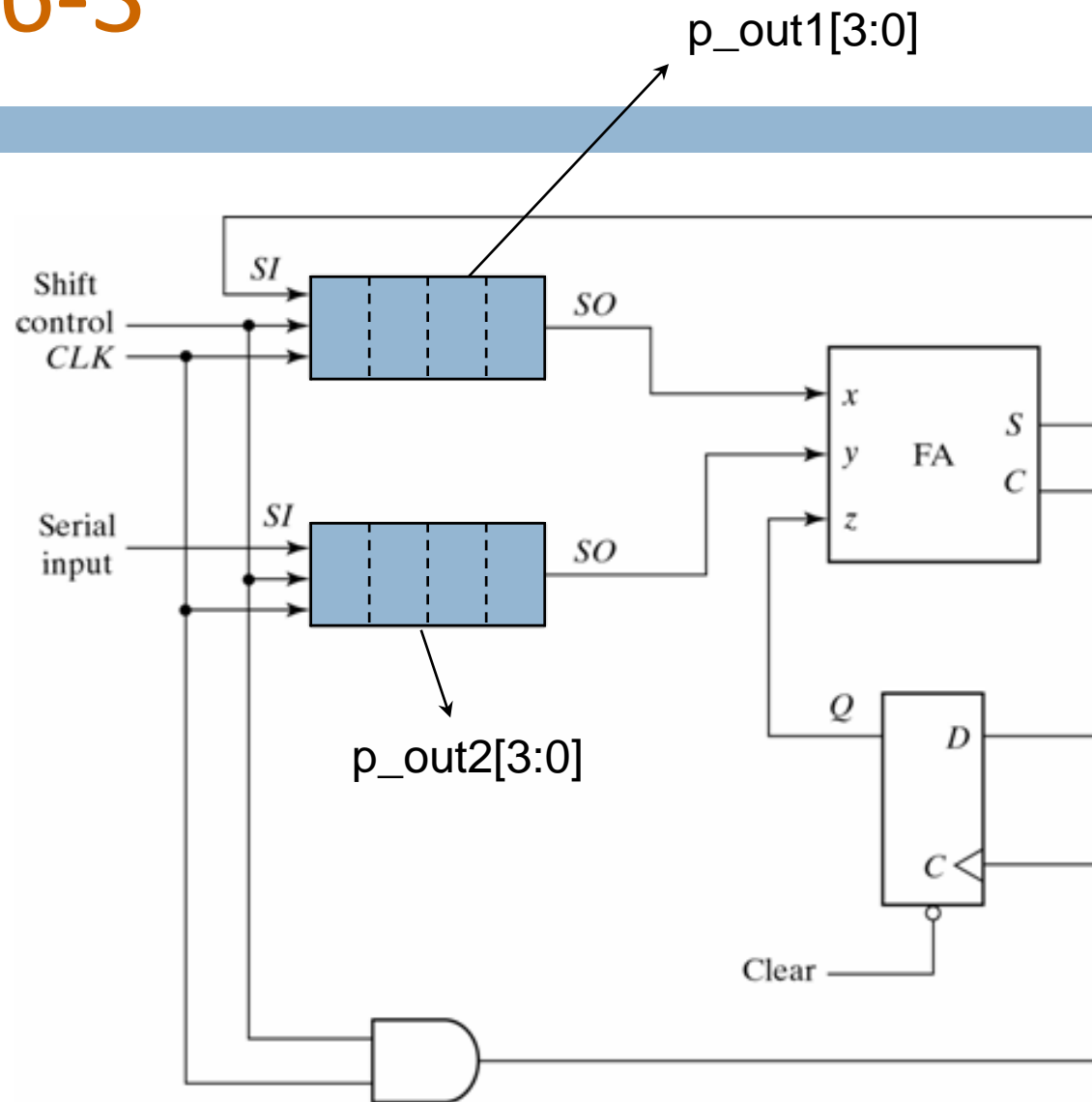
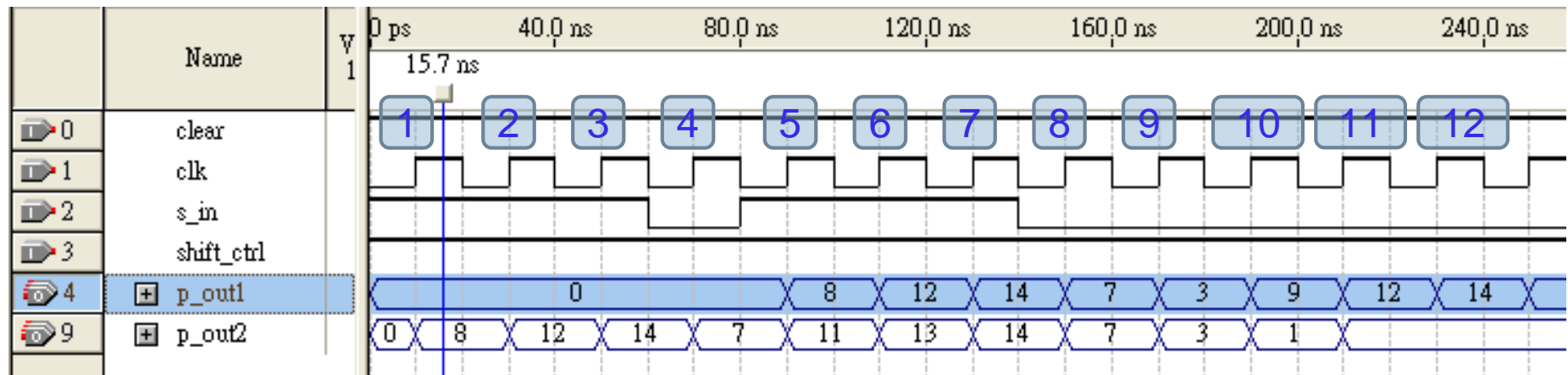


圖10.8

# LAB 6-3

27



# LAB 6-3

因為第三題比較複雜，同學可以使用我們的模板~

28

```
module lab6_3(clear,clk,p_out1,p_out2,s_in,shift_ctrl);  
input clear,clk,s_in,shift_ctrl;  
output [3:0]p_out1,p_out2;  
wire FA_C,FA_z,FA_S;
```

```
endmodule
```

(clk 正緣觸發 ; clear 負緣觸發)

```
module D_FF(clk,clear,D,Q);  
input clear;  
input clk,D;  
output reg Q;
```

```
endmodule
```

```
module FA(A, B, C0,S,C);  
input A, B, C0;  
output S, C;
```

```
endmodule
```

```
module shift(s_in,shift_ctrl,clk,p_out);  
input s_in;  
input shift_ctrl,clk;  
output [3:0] p_out;
```

```
endmodule
```

# LAB 6

29

下課前繳交至moodle：

上傳verilog.v

波形截圖