

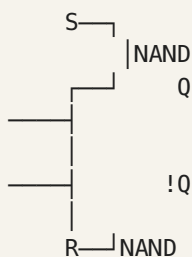
第 3 章 8：時序邏輯 (Sequential Logic)

時序邏輯有「記憶」，輸出依當前輸入 + 過去狀態。是 CPU、計數器、狀態機的基礎。

一、Latch 與 Flip-Flop

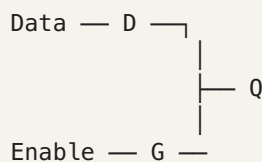
SR Latch (最簡單的記憶元件)

由兩個 NOR 或 NAND 互鎖：



S	R	Q (下一狀態)
0	0	不變
0	1	0 (reset)
1	0	1 (set)
1	1	不允許

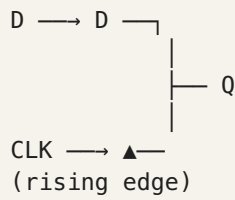
D Latch (電平觸發)



當 $G=1$ ：Q 跟隨 D 當 $G=0$ ：Q 鎖住

D Flip-Flop (邊緣觸發)

最常用的記憶元件，只在時鐘邊緣取樣：



只在 CLK 上升沿 (rising edge) 讀取 D，其他時間 Q 保持。

JK Flip-Flop

更通用的觸發器：

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	!Q(t) (toggle)

T Flip-Flop (Toggle)

T = 1 時翻轉，T = 0 時保持。用於分頻器 ($\div 2$)。

Master-Slave Flip-Flop

兩個 latch 串聯，避免「透通」 (race-around)。現代 D-FF 都是 Master-Slave 結構。

二、時序電路的關鍵時序參數

1. Setup time (建立時間)

CLK 邊緣前，D 必須穩定的時間。

2. Hold time (保持時間)

CLK 邊緣後，D 必須保持的時間。

3. Clock-to-Q (傳播延遲)

CLK 邊緣到 Q 出現新值的時間。

違反 Setup/Hold 的後果

訊號可能進入「亞穩態」 (Metastable)：- Q 介於 0 與 1 之間 - 持續時間不定 (μs - ms 級) - 邏輯誤判

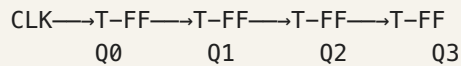
解決：跨時鐘域

不同時鐘域的訊號要透過 同步器 (兩級 D-FF) 才能安全使用。

三、計數器 (Counter)

非同步計數器 (Ripple Counter)

T-FF 串聯，每級 $\div 2$ ：



簡單但有累積延遲，最大頻率受限。

同步計數器

所有 FF 同時被一個時鐘觸發：



無累積延遲，適合高頻。

經典計數器 IC

- 74HC161：4-bit 同步計數器，可預載
- 74HC393：雙 4-bit 計數器
- 74HC4040：12-bit 計數器

四、暫存器 (Register)

並列暫存器 (Parallel Register)

n 個 D-FF 共用時鐘，存 n 位資料。

74HC273：8-bit register

移位暫存器 (Shift Register)

每個 CLK 邊緣資料移動一位：

In — [D-FF] — [D-FF] — [D-FF] — [D-FF] — Out
Q0 Q1 Q2 Q3

應用

- 串並轉換：UART、SPI 接收端
- 並串轉換：UART 傳送端
- 延遲線
- PRBS（偽隨機序列）產生

經典 IC

- 74HC164：SIPO (Serial In Parallel Out)
- 74HC165：PISO (Parallel In Serial Out)
- 74HC595：常用於擴展 GPIO (SPI 介面)

五、有限狀態機 (FSM, Finite State Machine)

數位系統的「演算法表達」。

兩種型態

Mealy 機

輸出依「狀態 + 當前輸入」。

Moore 機

輸出只依「狀態」。

設計流程

1. 畫狀態圖 (State Diagram)
2. 列狀態轉移表
3. 編碼狀態 (二進位、One-Hot、Gray code)
4. 推導 next-state 邏輯
5. 寫 HDL 或畫電路圖

Verilog 範例：簡單 FSM

```

module fsm_example (
    input clk, rst, x,
    output reg y
);
    reg [1:0] state, next_state;

    parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10;

    always @(posedge clk or posedge rst)
        if (rst) state <= S0;
        else state <= next_state;

    always @(*) begin
        case (state)
            S0: next_state = x ? S1 : S0;
            S1: next_state = x ? S2 : S0;
            S2: next_state = S0;
            default: next_state = S0;
        endcase
    end

    always @(*) y = (state == S2);
endmodule

```

六、計時與分頻

分頻器

用 T-FF 串聯：每級頻率減半。8 個 T-FF $\rightarrow \div 256$

計數分頻

要 $\div N$ (不是 2 的次方)：

設計同步計數器，當計到 $N-1$ 時 reset。

PLL 倍頻

從低頻產生高頻（見 04-振盪器章）。

七、時鐘樹 (Clock Tree)

複雜系統有大量 FF 需要同時被時鐘觸發：

Skew (時鐘偏斜)

不同 FF 收到時鐘的時間不同：- 走線長度差異 - 緩衝器延遲差 - 環境影響

Clock Tree Synthesis (CTS)

EDA 工具自動插入緩衝器，使 skew 最小。

Jitter (抖動)

時鐘週期的隨機變化。影響：高速通訊 (SerDes)、ADC 取樣。

八、常見時序電路範例

1. 紅綠燈控制器

3 個狀態（紅、黃、綠），FSM 設計。配合計數器計時切換。

2. PWM 產生器

計數器 + 比較器：- 計數器從 0 計到 N - 與 duty value 比較，輸出 0/1

3. UART 接收器

- 偵測 start bit (下降沿)
- 開始計時
- 取樣中央時刻
- 移位暫存器收 8 位
- 檢查 stop bit
- 輸出 byte

4. SPI 控制器

- 8-bit 移位暫存器
 - 時鐘控制
 - 片選邏輯
 - DMA 整合
-

九、本章總結

✅ 您應該掌握：

- [x] Latch 與 Flip-Flop 的差別
 - [x] D/JK/T Flip-Flop
 - [x] Setup/Hold/Clock-to-Q 時序參數
 - [x] 亞穩態與同步器
 - [x] 計數器、暫存器、移位暫存器
 - [x] FSM（有限狀態機）設計
 - [x] 時鐘樹、Skew、Jitter
-

下一章

09-???.md 一 各種記憶體技術。