

國立清華大學 電機工程學系  
實作專題研究成果報告

A Fast-Locking All-Digital Fractional-Ratio  
Multiplying DLL

可實現快速鎖定的全數位分數比乘頻延遲  
鎖定迴路設計

專題領域：系統領域

組 別：A394

指導教授：黃錫瑜教授

組員姓名：林士登、陳立珩

研究期間：113年1月16日 至 113 年5月1日 止，共 3.5個月

## Abstract

In today's increasingly complex chip design architectures, the timing circuits used to stabilize the system have become increasingly crucial. Although traditional high-speed Phase-Locked Loops (PLLs) have long been a staple for circuit stability, their inherent jitter performance and longer locking times limit their use in more advanced applications. This has prompted researchers to look for alternatives, and the all-digital Multiplying Delay-Locked Loop (MDLL) has emerged as a result. With its superior jitter performance and fast locking capabilities, MDLL is gradually replacing PLLs in high-performance systems, although this circuit operation mode has its own non-ideal characteristics.

The focus of this project is to restore an innovative all-digital Fractional Ratio Multiplying Delay-Locked Loop (FMDLL), a new architecture that addresses many challenges in the MDLL system, particularly in achieving end-to-end device frequency synthesis. The FMDLL, through a three-mode operation control logic and an innovative phase detection structure, precisely controls timing digitally, achieving finer frequency adjustments and faster response times to eliminate internal non-ideal phase differences. The FMDLL employs a novel selection logic, combined with an efficient phase detector and a harmonic lock detector, to ensure stable clock output even under complex frequency transition conditions. Additionally, using a binary search algorithm's Successive Approximation Register (SAR), the locking process is significantly shortened, enhancing the system's overall responsiveness.

In summary, this project utilizes the TSMC 90nm CLN90G process standard cell library and the Cell Based Design Flow method to recreate the new type of FMDLL proposed in the paper [1], providing a multiplication ratio of  $N/M$  ( $N > M$ ). Considering the existing process technology and circuit complexity, specific  $N$  values that destabilize the divider are excluded, along with many different but repetitive fractional ratios of  $N$  and  $M$  values. Therefore, the design specifications refer to paper [1] set as  $M[1:0] = \{1, 2, 3\}$ ,  $N[3:0] = \{1, 4, 5, 8, 10\}$ , not only improving jitter control and locking times over traditional technologies but also opening new possibilities in meeting dynamic frequency scaling and precise frequency synthesis requirements.

## 摘要

在現今的晶片設計架構系統愈來愈複雜的情況下，用來穩定系統的時序電路也變得日益重要。傳統的高速時脈鎖相迴路（PLL）雖然長期作為支持電路穩定的存在，但其固有的抖動表現和較長的鎖定時間限制了其他更為先進的應用。這促使研究者尋找替代方案，全數位乘法延遲鎖定迴路（MDLL）因此應運而生，其卓越的抖動性能及快速的鎖定能力，使其在高性能系統中逐漸取代鎖相迴路，但這種電路運作方式卻存在其它非理想特性。

本專題的重點在於還原一種創新的全數位分數比乘頻延遲鎖定迴路（FMDLL），這一新型架構解決了MDLL系統中存在的諸多挑戰，特別是在實現端對端裝置的頻率合成上。FMDLL 藉由三個模式操作控制邏輯和創新的相位檢測結構，以數位方式精確控制時序，從而實現更精細的頻率調整和更迅速的反應時間，以消除內部非理想的相位差。FMDLL採用的是一種新穎的選擇邏輯，結合了高效的相位檢測器與和諧波鎖定檢測器，以保證即使在複雜的頻率轉換條件下也能維持穩定的時鐘輸出。此外，利用二進制搜索算法的逐漸逼近暫存器（SAR），大幅縮短了鎖定過程，提高了系統整體的響應速度。

總結而言，本專題使用 TSMC 90 奈米 CLN90G 製程的標準元件庫，利用 Cell Based Design Flow 設計方法試圖還原研究文獻 [1] 所提出的新型 FMDLL，可提供電路  $N/M$  ( $N>M$ ) 的乘頻倍率，考量到現有製程技術與電路複雜度，排除特定  $N$  值對除頻器的不穩定性以及許多不同  $N$ 、 $M$  值但重複的分數比，因此設計時規格參考文獻 [1] 設定為  $M[1:0] = \{1, 2, 3\}$ 、 $N[3:0] = \{1, 4, 5, 8, 10\}$ ，不僅在抖動控制和鎖定時間上優於傳統技術，還在滿足動態頻率調節和精密頻率合成需求上開闢了新的可能。

# 目錄

<b>1. Introduction</b>	1
<b>2. Research Methodology</b>	2
2.1 Proposed All-digital SAR-Based FMDLL Architecture	2
2.2 Internal Circuit Modules and Operating Mechanism	3
2.2.1 Phase Tracking Controller (PTC)	3
2.2.2 Harmonic Lock Detector (HLD)	4
2.2.3 Select Logic	5
2.2.4 Digital Control Delay Line (DCDL)	6
<b>3. Experimental Results</b>	7
3.1 Pre-layout Simulation	7
3.2 Post-layout Simulation	9
3.3 Layout vs. Gate-Level-Synthesis	12
<b>4. Conclusion</b>	14
<b>5. Reference</b>	14
<b>6. Plan Management and Teamwork</b>	15

# 1. Introduction

## 1.1 Research Background

動態頻率調節為基礎的電源管理在低功耗處理器和記憶體設計中的關鍵元素，而使用數位晶片時脈產生器或乘頻器時，能提供具有快速重新同步和去偏能力的多變系統時脈頻率。最近，已引入能提供頻率乘法的循環或乘法延遲鎖定迴路（MDLL），它們比相位鎖定迴路（PLL）提供更優越的抖動性能。一個典型的 MDLL 通過定期開啟環形延遲線並注入乾淨的參考輸入時脈訊號邊緣來減少抖動累積。但缺點是在於傳統的 MDLL 只能提供整數比（ $=N$ ）的頻率乘法。為了解決這個問題並擴展其應用，研究文獻提出了一種能提供分數比（ $=N/M$ ）頻率乘法的新 MDLL 架構。此外，許多實際問題如快速鎖定需求與和諧波鎖定問題也未被考慮。在許多 MDLL 架構中，當乘法輸出時脈訊號的初始頻率低於期望頻率時，很容易發生和諧波鎖定（Harmonic Lock）。在這種情況下，輸出時脈訊號會鎖定到輸入時脈訊號的和諧波參考邊緣，導致非理想的輸出頻率或鎖定失效。

為了改善和諧波鎖定並提供快速鎖定功能，我們的專題研究參考了文獻 [1] 中的電路架構並嘗試還原其電路功能，利用以下幾種創新電路模塊來解決 MDLL 鎖定時間長與和諧波鎖定的問題。

1. 逐漸逼近暫存器 (Successive Approximation Register)：利用類似二元搜索的演算法搭配新的數位延遲導線 (DCDL)，來達到快速鎖定與大範圍頻率的動作。
2. 和諧波偵測器 (Harmonic Lock Detector)：這個模組提供偵測內部運作模式選擇訊號的變化，偵測邏輯錯誤並改善其錯誤。
3. 相位追蹤與控制電路 (Phase Tracking Controller)：將上述的 SAR 與 HLD 結合並搭配相位偵測電路輸出控制碼，控制延遲導線的延遲時間。
4. 邏輯選擇模組 (Select Logic)：利用內部的時脈訊號迴授做出邏輯判斷，並利用三個模式的切換來達成輸出時鐘訊號的穩定。

在本次專題研究中，全數位分數比乘法延遲鎖定迴路（FMDLL）不僅在理論上具有創新性，而且在實驗中也證明了其實用性和效率。通過新穎的和諧波鎖定檢測器和先進的相位追蹤控制系統，此架構有效地克服了傳統 MDLL 的局限，提供了更快的鎖定速度和更高的操作靈活性，使其特別適用於要求高速與低功耗的現代電子系統。接下來的部分將詳細介紹系統架構、鎖定過程、詳細的電路實現與實驗步驟和結果，最後總結本研究的成果。



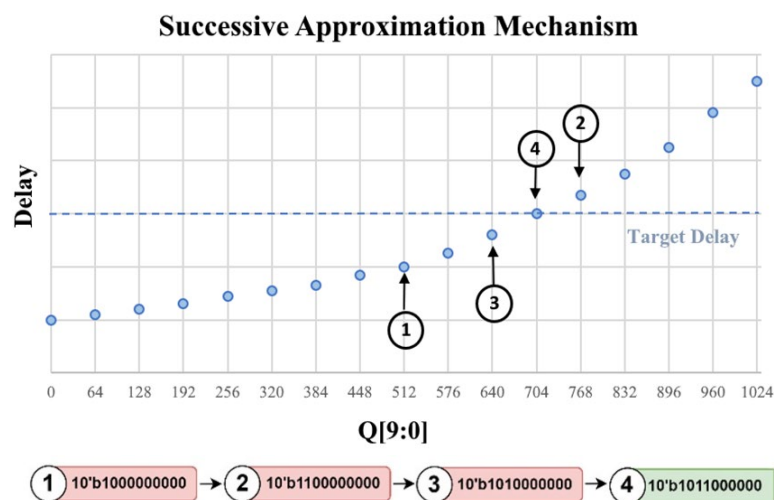
## 2.2 Internal Circuit Modules and Operating Mechanism

### 2.2.1 Phase Tracking Controller (PTC)

PTC 如 Fig.2-1 之架構使用 10-bit SAR 去實現 Fig.2-2 二元搜尋演算法之電路，以方便尋找適合的延遲控制碼  $Q[9:0]$  控制輸出乘頻訊號之週期，相較於線性搜尋法可以達到更高的頻率和更低的鎖定時間，而此 SAR 與除以 4 的時脈分頻器產生的  $CLK_4$  信號同步，若除數  $M$  是整數 1，使得  $M[1:0]=[0\ 1]$ ，則  $CLK_4$  信號由外部時脈  $CLK_{EXT}$  除以 4 產生；若  $M$  大於 1，則  $CLK_4$  由  $DIV_M$  信號除以 4 產生。同理，當  $M$  不等於 1 時， $CLK_2$  信號由  $DIV_M$  信號除以 2 產生。

另外， $N[2:0]$  信號用於可輸入的乘法因子，而  $M[1:0]$  信號用於可輸入的除法因子，這種基於 SAR 的二元搜尋演算法只需要  $4 \times k \times M$  個 reference clock ( $k$  為  $Q$  的位元數) 週期來進行鎖定。舉例來說，若  $k = 10$  且  $N/M = 5/3$ ，則以 SAR 建構的 FMDLL 的鎖定時間只需  $4 \times 10 \times 3 = 120$  個週期，當二元搜尋演算法完成後，SAR 輸出的  $Q[9:6]$  傳給 4 對 16 解碼器轉譯生成溫度計碼 (Thermometer Code)，給粗糙延遲導線 (Coarse Delay Line) 使用並產生延遲信號，而  $Q[5:3]$  和  $Q[2:0]$  分別傳給兩個 3 對 8 解碼器作轉譯生成另一組溫度計碼，給精細延遲導線 (Fine Delay Line) 使用，並同時在封閉迴路持續追蹤 PVT 和環境變化，並且減少 jitter。

這裡使用的相位偵測器 (Phase Detector) 偵測  $CLK_{EXT}$  的  $(M+1)$  次正緣觸發上升是領先還是落後於  $CLK_{OUT}$  的  $(N+1)$  次正緣觸發上升來產生相位比較訊號  $COMP$ 。舉例來說，若  $CLK_{EXT}$  的  $(M+1)$  次上升領先於  $CLK_{OUT}$  的  $(N+1)$  次上升，相位偵測器會產生一個 Lag 訊號，代表  $CLK_{OUT}$  目前落後於  $CLK_{EXT}$ ，我們令  $COMP$  為 High，此時需要將延遲時間縮短使  $CLK_{OUT}$  訊號更容易趕上  $CLK_{EXT}$  訊號；若  $CLK_{EXT}$  的  $(M+1)$  次上升落後於  $CLK_{OUT}$  的  $(N+1)$  次上升，相位偵測器會產生一個 Lead 訊號，代表  $CLK_{OUT}$  目前領先於  $CLK_{EXT}$ ，我們令  $COMP$  為 Low，此時需要將延遲時間增長使  $CLK_{EXT}$  訊號更容易趕上  $CLK_{OUT}$  訊號。



**Fig.2-2 逐漸逼近演算法 (Successive Approximation)**

## 2.2.2 Harmonic Lock Detector (HLD)

儘管使用以 SAR 建構的 FMDLL 可以帶來更快的鎖定時間，但存在著風險，因為在操作開始時，當控制碼 Q 的最高有效位元 (MSB) 發生變化時，二元搜尋法可能會導致輸出頻率和相位出現突然大變化，如果數位延遲導線 (DCDL) 的延遲過大，則 FMDLL 容易鎖定到 CLK<sub>OUT</sub> 的上升邊緣，導致錯誤的 N/M 和非期望的輸出頻率，這種情況類似於傳統 MDLL 中的和諧波鎖定 (Harmonic Lock) 問題。

為了避免此錯誤鎖定問題，在 PTC 中設計一個 Harmonic Lock Detector (HLD) 來偵測錯誤並改善。因此 FMDLL 在操作過程中無須擔心是否因 DCDL 的延遲過大造成的相位比較錯誤，而可大幅提升操作頻率。Fig.2-3 圖為 Harmonic Lock Detector 的架構，Harmonic Lock 觀察模式選擇訊號 Sel[0] 的變化，如果 Sel[0] 在原本應該升起的位置沒有訊號，會使得 HLD 內部的導線 HLD<sub>1</sub> 無法被給定 Low 值，造成強制重整相位偵測器之訊號 Reset<sub>PD</sub> 升起，因此使得相位比較訊號 COMP 維持 Low 值，DCDL 的延遲下降，直到 CLK<sub>4</sub> 升起且同時 Sel[0] 升起時，代表電路功能回歸正常狀態，Reset<sub>PD</sub> 才變成 Low，相位恢復到可比較的狀態，詳細情形如 Fig.2-4 圖說明。

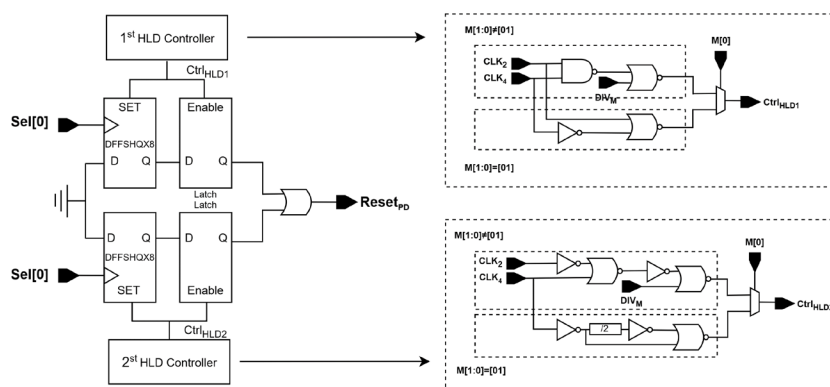


Fig.2-3 和諧波鎖定偵測器 (HLD) 內部電路

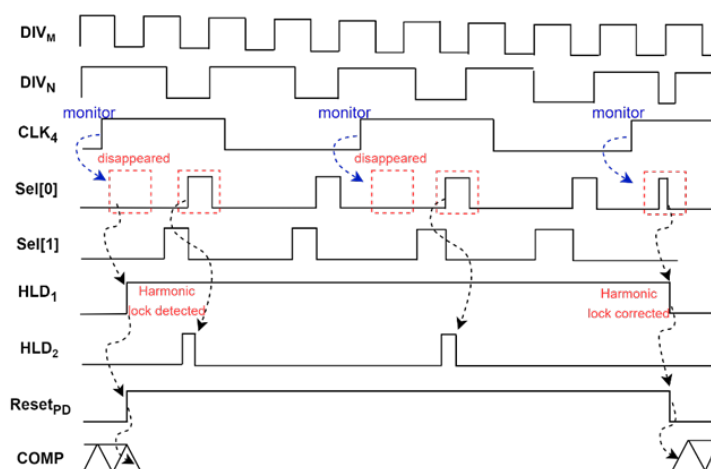


Fig.2-4 Harmonic Lock 問題以及 HLD 運作原理



### 2.2.3 Select Logic

Fig.2-6 圖中的 $\Delta T$  為  $N/M=8/3$  要達到鎖定條件所需的 DCDL 延遲，此延遲時間比輸入的  $CLK_{EXT}$  週期還要長(在圖中表示為  $t_{cycle}$ )， $t_{initial}$  為最初 DCDL 的延遲。

當 FMDLL 開始運作時，會進入 RO (Ring Oscillator) 模式，此時令  $Sel[1:0]=[0\ 0]$ 。如果一開始 DCDL 的延遲時間太短，也就是  $\Delta T > t_{cycle}$ ，那麼當  $CLK_{OUT}$  震盪到第 N 個週期時，使得  $DIV_N$  下降到 Low，此時  $DIV_M$  為 High， $Sel[1]$  會變成 High，接著進入 SI (Supply Injection) 模式，此時令  $Sel[1:0]=[1\ 0]$ 。在 SI 模式時，此時多工器會輸入 High 進延遲路徑，直到第 M 個  $CLK_{EXT}$  抵達，此行為可大幅增加延遲時間。在  $CLK_{EXT}$  抵達後， $DIV_M$  會變成 Low，會讓  $Sel[1]$  變成 Low、 $Sel[0]$  變成 High，此時令  $Sel[1:0]=[0\ 1]$ ，進入 RI (Reference injection) 模式，此模式中會讓多工器會輸入  $CLK_{EXT}$  進延遲路徑，這樣可以消除在 RO 震盪中所累積的 jitter。而當第 M+1 個  $CLK_{EXT}$  抵達後， $DIV_M$  會再次回到 High，讓  $Sel[0]$  變成 Low，重新進入 RO 模式。

在 PTC 中的 10-bits SAR 會根據 PD 比較第 M+1 個  $CLK_{EXT}$  和第 N+1 個  $CLK_{OUT}$  的領先落後情形增加或減少 Q 值，Fig.2-7 說明當 DCDL 延遲不斷增加，使得第 M 個  $CLK_{EXT}$  的升起比第 N 個  $CLK_{OUT}$  的升起還早時，此種情況就不會進入到 SI 模式，而是會週期性的不斷重複 RO 和 RI 模式，如同傳統的 MDLL 一樣，當第 M+1 個  $CLK_{EXT}$  的升起對齊第 N+1 個  $CLK_{OUT}$  的升起，代表完成  $N/M$  的頻率乘法，詳細的 Select Logic 運作流程圖在 Fig.2-5 呈現。

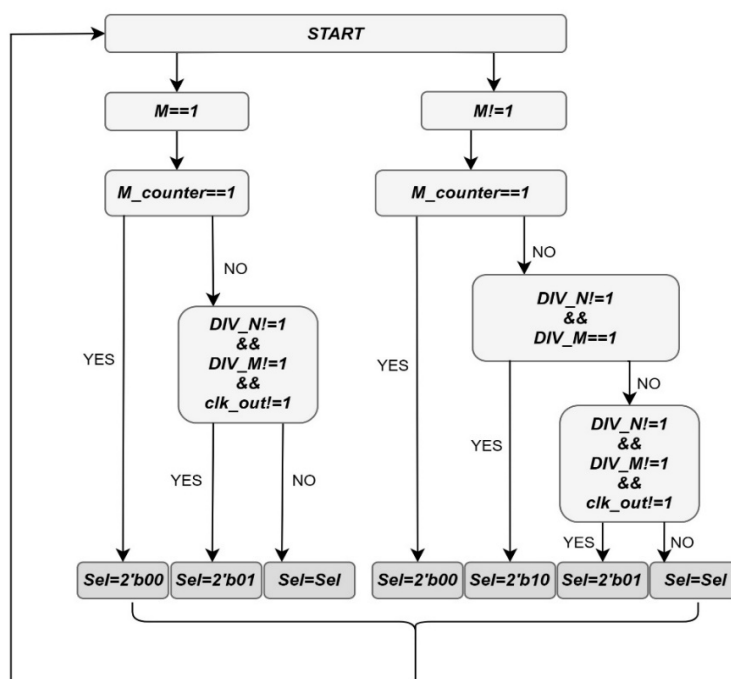
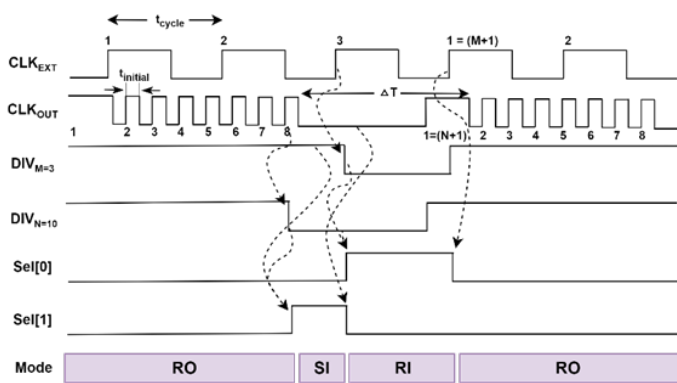
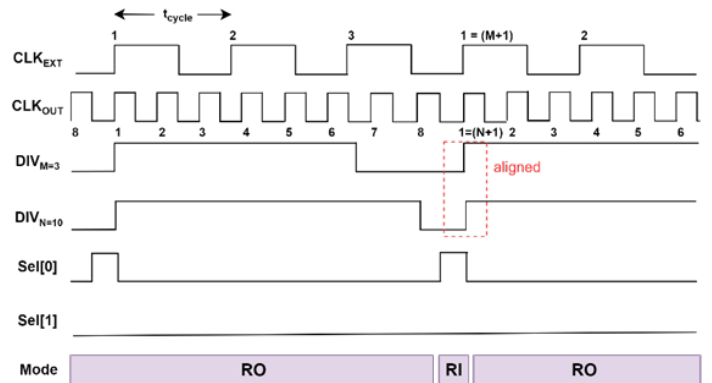


Fig.2-5 Select Logic flow chart



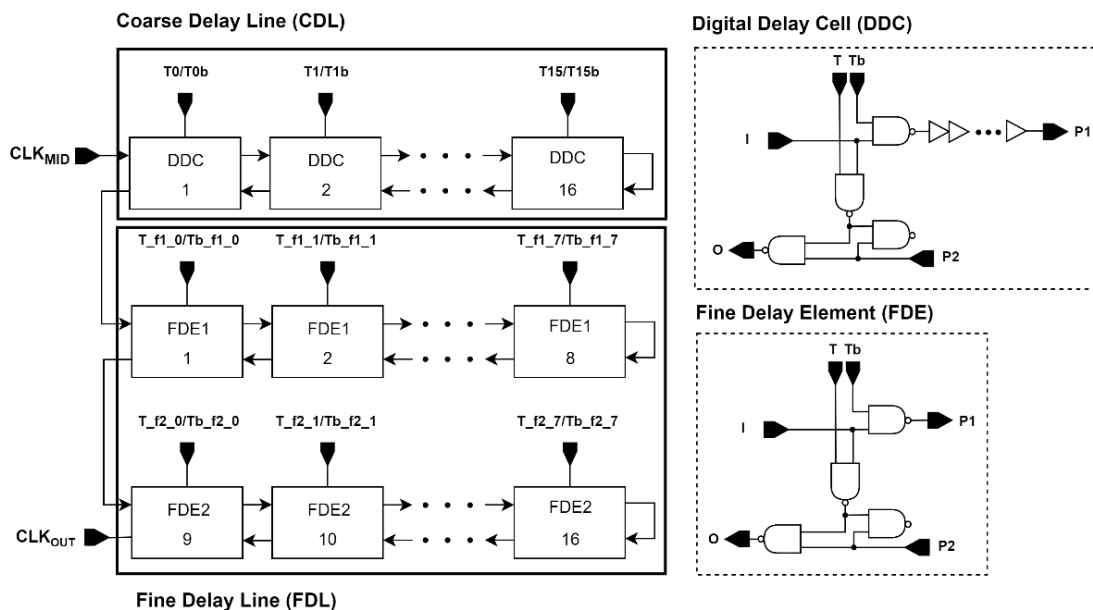
**Fig.2-6** 詳細模式切換來達成  $N/M=8/3$  的頻率



**Fig.2-7**  $N/M=8/3$  的頻率鎖定後的情形

## 2.2.4 Digital Control Delay Line (DCDL)

Fig.2-8 圖說明了此 FMDLL 使用的 DCDL 延遲架構。CDL (Coarse Delay Line) 由 16 個 DDC (Digital Delay Cell) 所組成，而每個 DDC 是由四個 NAND 邏輯閘和些許的 Buffer 構成，延遲多寡的控制是由  $T[15:0] / T_b[15:0]$  決定，由於  $Q[9:6]$  已經經過 4 對 16 解碼器轉譯變成溫度計碼，所以一個  $Q[9:6]$  只會產生一條延遲路徑，當  $Q[9:6]$  越大，訊號需要經過更多的 NAND 邏輯閘，產生更大的延遲。而我們使用的 FDL (Fine Delay Line) 與 CDL 架構基本一致，且由 FDE (Fine Delay Element) 組成，而 FDE 與 DDC 不同的地方是少了 Buffer，所以產生的延遲會比較小，為了區別 FDE1 和 FDE2，我們使用不同尺寸的 NAND 邏輯閘來達到 FDE1 延遲比 FDE2 還大的效果，而延遲多寡控制是由  $T_{f1}[7:0] / T_{b_{f1}}[7:0]$  ( $T_{f2}[7:0] / T_{b_{f2}}[7:0]$ ) 決定。



**Fig.2-8** DCDL延遲架構

### 3. Experimental Results

本專題使用的是 Cell-Based Design Flow 方法設計，大致分為三個部分來執行，首先利用 Verilog 語言 RTL coding，將我們的模組都設計完後，要去執行邏輯合成，接下來要拿合成過後的 Gate-Level Netlist 去模擬並跑 pre-layout 之波形圖，若發現過程中波形圖出現不合預期的功能或錯誤，要回去檢查並改善自己設計的 RTL code。最終部分為執行寫好的 APR (Auto Placement and Routing) script，並檢查 layout 是否有接線或是短路問題並試著改善，再將合成出來的 netlist 去跑 post-layout 之波形圖，最後檢查是否與期望之波型相同。Fig.3-1 為專題進行期間整個 Cell-Based Design Flow 的流程。

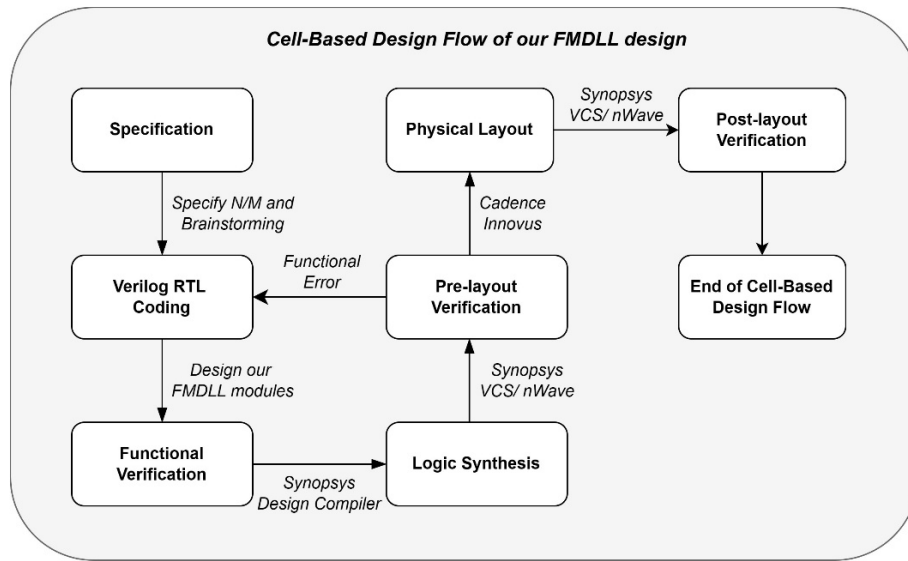


Fig.3-1 Cell-Based Design Flow chart

#### 3.1 Pre-layout Simulation

以下是 RTL code 在經過 Design Compiler 邏輯合成成為 Gate-Level Description 之後的 netlist，再透過 VCS 與 nWave 模擬 pre-sim 之波形圖，以下將列舉三種 N/M 分數比的波形圖為例。

1.  $N/M=10/3$ ,  $T_{clk\_ext} = 30ns$

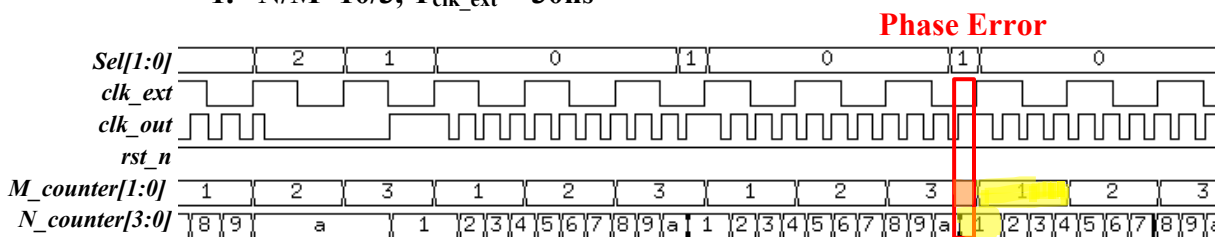
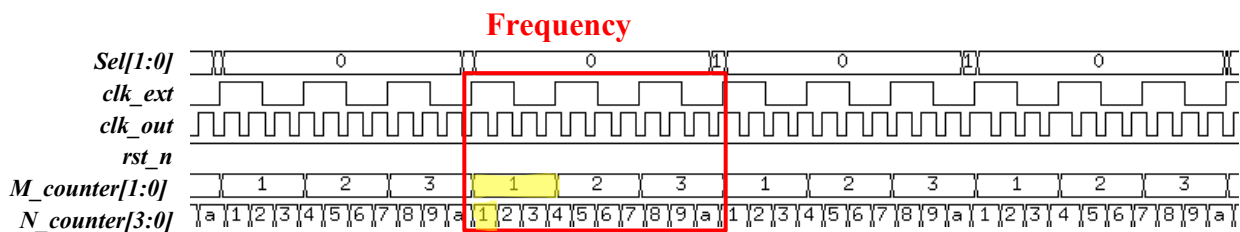


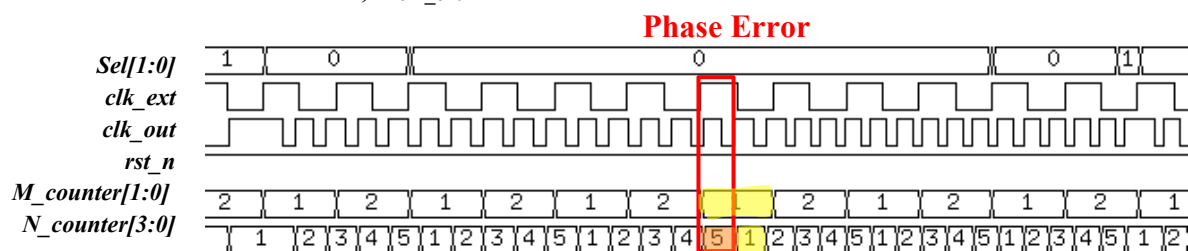
Fig.3-2  $f_{clk\_out} = (10/3) \times f_{clk\_ext}$  尚未鎖定



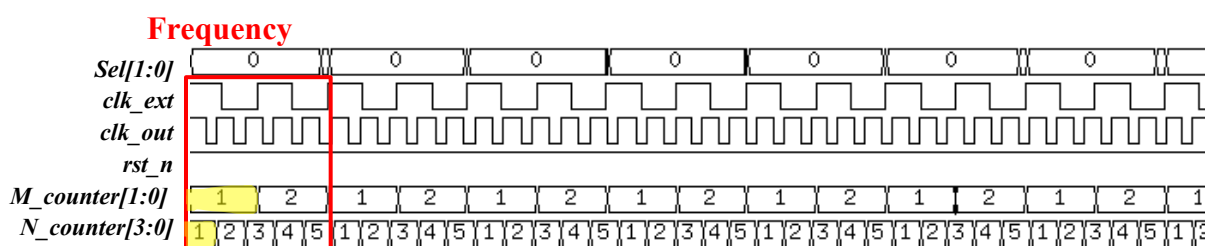
**Fig.3-3**  $f_{clk\_out} = (10/3) \times f_{clk\_ext}$  頻率已鎖定

取  $N/M=10/3$  且輸入時脈訊號  $clk\_ext$  週期為 30ns 情況下，換算之輸入頻率為 33.33MHz，可以觀察到在 Fig.3-2 還在搜尋目標延遲而存在相位差，後來在內部電路延遲愈條愈精細的情況下 Select Logic 模式切換已剩下 RO 與 RI 模式在進行微調，最後結果如 Fig.3-3 之頻率鎖定，而鎖定時間為 3787.567ns，再扣除一開始電路運行至穩定之 Reset 時間 200ns，可以得知總共花費鎖定時間為輸入訊號之 119.58 個週期，與理想鎖定時間  $4 \times k \times M = 4 \times 3 \times 10 = 120$  個輸入訊號週期差不多，最終  $f_{clk\_out} = f_{clk\_ext} \times (10/3) = 111.11\text{MHz}$ 。

## 2. $N/M=5/2$ , $T_{clk\_ext} = 24\text{ns}$



**Fig.3-4**  $f_{clk\_out} = (5/2) \times f_{clk\_ext}$  尚未鎖定



**Fig.3-5**  $f_{clk\_out} = (5/2) \times f_{clk\_ext}$  頻率已鎖定

取  $N/M=5/2$  且輸入時脈訊號  $clk\_ext$  週期為 24ns 情況下，頻率約為 41.67 MHz，觀察 Fig.3-4 在電路剛啟動時還在利用 SAR 尋找目標延遲，此時都還存在著相位差，直到控制碼 Q 迭代到最後一個位元(LSB)後，相位鎖定至  $N/M=5/2$  如 Fig.3-5，而經過的時間為 1990.885ns，扣掉系統穩定時間 150ns，共花了 76.5 個週期達到鎖定狀態，而此次鎖定時間滿足理想週期  $4 \times k \times M = 4 \times 10 \times 2 = 80$  個週期的時間。最終鎖定頻率為  $f_{clk\_out} = f_{clk\_ext} \times (5/2) = 104.17\text{MHz}$ 。

### 3. $N/M=8/1$ , $T_{clk\_ext} = 50ns$

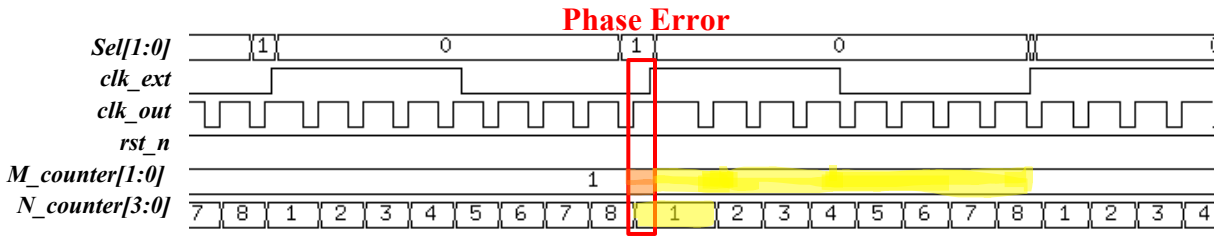


Fig.3-6  $f_{clk\_out} = (8/1) \times f_{clk\_ext}$  尚未鎖定

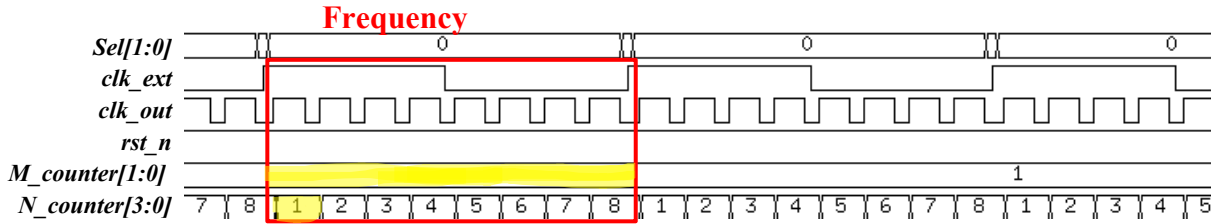


Fig.3-7  $f_{clk\_out} = (8/1) \times f_{clk\_ext}$  頻率已鎖定

取  $N/M=8/1$  且輸入時脈訊號  $clk\_ext$  週期為 50ns 情況下，頻率為 20 MHz，與上述兩個例子相同，Fig.3-6 至 Fig.3-7 由存在相位差而後達到頻率鎖定，扣除電路穩定時間後經過的鎖定時間為 2000.221ns，因此共經過了約 40 個輸入週期，與理想鎖定時間  $4 \times k \times M = 4 \times 10 \times 1 = 40$  個週期差不多，最終鎖定頻率為  $f_{clk\_out} = f_{clk\_ext} \times (8/1) = 160$  MHz。

確認合成過後的 FMDLL netlist 之功能正常後，接下來要將 netlist 上傳至 Innovus 執行自動佈局以及繞線 (Auto Placement and Routing)，產生 layout 後儲存 layout netlist 並執行 post-layout simulation。APR layout 與 Gate-Level-Synthesis 之細部內容與相關比較將會在 3.3 節詳述。

## 3.2 Post-layout Simulation

經過 APR 自動合成 layout 後，產生了最終的 post-layout netlist 檔案，在這個小節，會將如同以上 pre-layout 模擬所使用的例子之波形圖呈現出來以及比較與 pre-layout 模擬的不同之處。

### 1. $N/M=10/3$ , $T_{clk\_ext} = 30ns$

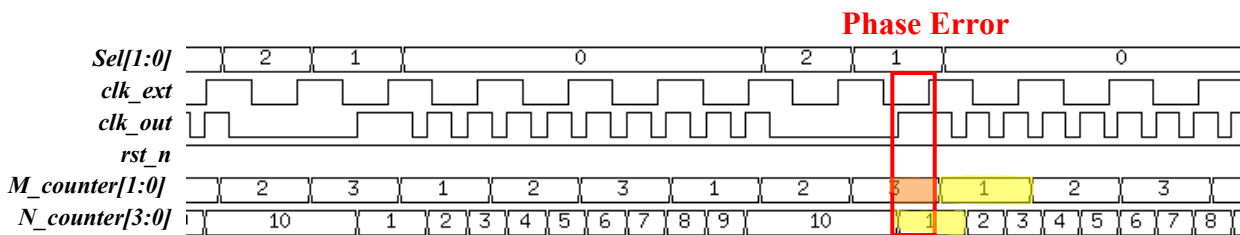


Fig.3-8  $f_{clk\_out} = (10/3) \times f_{clk\_ext}$  尚未鎖定



由 post-layout 模擬可以觀察到波形圖和電路的運作與 pre-layout 模擬大同小異，不同之處在於因為 pre-layout 尚未考慮到 layout 上的金屬導線延遲、寄生電阻電容效應等等非理想特性，而是只使用製程檔中 standard cell delay 模型，因此在 layout 佈局過後，每顆邏輯閘延遲都會相較於 pre-layout 還來的高，導致 pre-layout 模擬與 post-layout 模擬之可操作頻率範圍不相同。Table 3-1 與 Table 3-2 紀錄在 pre-layout 與 post-layout 針對不同分數比 N/M 之支援頻率範圍與輸出頻率範圍的比較。

**Table 3-1**  
FMDLL Input Operating Frequency Range (Pre-Layout vs. Post-Layout)

Pre-Layout (N/M)	Input Frequency (MHz)	Post-Layout (N/M)	Input Frequency (MHz)
10/3	18.52~52.08	10/3	16.67~33.33
8/3	23.26~60.98	8/3	15.63~25.00
5/3	35.97~87.72	5/3	22.73~33.33
4/3	45.45~88.50	4/3	26.32~50.00
10/2	12.35~28.09	10/2	7.69~12.18
8/2	15.38~28.74	8/2	1.00~27.78
5/2	24.39~45.45	5/2	15.63~25.00
4/2	30.30~56.18	4/2	20.00~27.78
10/1	10.00~16.13	10/1	3.85~11.11
8/1	10.64~21.28	8/1	5.00~14.29
5/1	12.50~33.33	5/1	7.69~14.29
4/1	15.15~41.67	4/1	9.62~14.71

**Table 3-2**  
FMDLL Output Frequency Range (Pre-Layout vs. Post-Layout)

Pre-Layout (N/M)	Output Frequency (MHz)	Post-Layout (N/M)	Output Frequency (MHz)
10/3	61.73~173.61	10/3	55.57~111.10
8/3	62.02~162.60	8/3	41.68~66.67
5/3	59.95~146.20	5/3	37.88~55.55
4/3	60.61~117.99	4/3	35.09~66.67
10/2	61.73~140.45	10/2	38.45~60.90
8/2	61.54~114.94	8/2	4.00~111.12
5/2	60.98~113.64	5/2	39.08~62.50
4/2	60.61~112.36	4/2	40.00~55.56
10/1	100.00~161.29	10/1	38.46~111.10
8/1	85.11~170.21	8/1	40.00~114.32
5/1	62.50~166.67	5/1	38.46~71.45
4/1	60.61~166.67	4/1	38.46~58.84

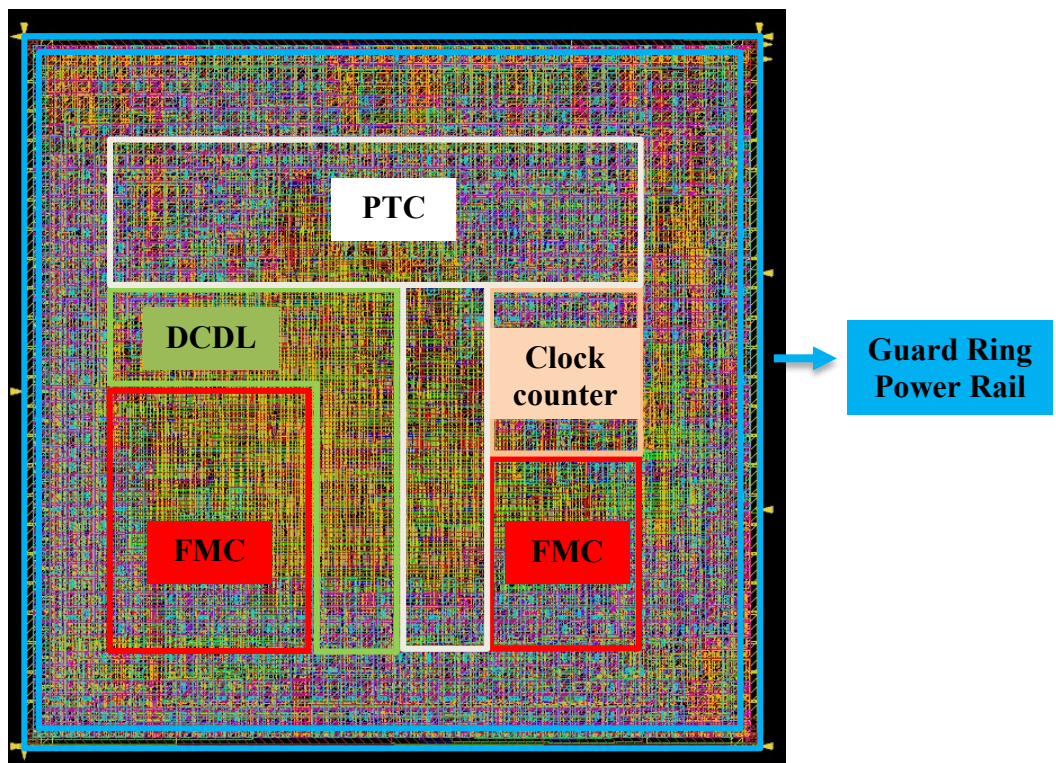


由 Table 3-1 與 Table 3-2 可以發現 pre-layout simulation 的支援頻率範圍很大，但 post-layout simulation 的支援頻率範圍稍小了一些，表示 layout 佈局的好壞影響也不容小覷，下表 Table 3-3 整理了 Table 3-1 與 Table 3-2 之輸入與輸出訊號的最大最小頻率範圍。

**Table 3-3**  
Maximum & Minimum Overall Frequency Range

	Input Frequency Range (MHz)	Output Frequency Range (MHz)
<b>Pre-Layout</b>	10.00~88.50	59.95~173.61
<b>Post-Layout</b>	1.00~50.00	4.00~114.32

### 3.3 Layout vs. Gate-Level-Synthesis



**Fig.3-14** Cell-Based FMDLL Layout

由 Fig.3-14 可以看到 FMDLL 佈局圖的模塊大致分布，環型電壓源  $V_{DD}$  在此製程下給予 1.0V，而 layout 中的 Clock\_counter 負責分別計算輸入時脈訊號與輸出時脈訊號從 1 到 M 次與 1 到 N 次的週期，提供設計時觀察訊號頻率是否鎖定的便利性。



下表 Table 3-4 將比較在 pre-layout 時 Gate-Level-Netlist 所估計使用的面積以及在 post-layout 時佈局的面積，也將比較它們的功率，並在 Table 3-5 與參考文獻 [1] 所研發的 FMDLL 進行表現比對。

**Table 3-4**  
Gate-Level-Netlist vs. Layout

	Gate-Level-Netlist	Layout
<b>Total Cell Area (um<sup>2</sup>)</b>	2554.27	3604.91
<b>Internal Power (mW)</b>	0.086941 (63.85%)	0.128593 (46.41%)
<b>Switching Power (mW)</b>	0.038836 (28.53%)	0.139035 (50.18%)
<b>Leakage Power (mW)</b>	0.010369 (7.62%)	0.009459 (3.41%)
<b>Total Power (mW)</b>	0.136146	0.277009

**Table 3-5**  
FMDLL Performance Comparison In Paper Reference [1]

	[1] TCAS-I	Our Work
<b>Architecture</b>	All Digital	All Digital
<b>Technology</b>	TSMC 65nm	TSMC 90nm
<b>Supply (V)</b>	1.0	1.0
<b>Output Freq. (GHz)</b>	0.7 – 2.0	0.004 – 0.114
<b>Multiplication Factor [N] / Division Factor [M]</b>	N={1, 4, 5, 8, 10} M={1, 2, 3}	N={1, 4, 5, 8, 10} M={1, 2, 3}
<b>Peak-to-Peak jitter (ps) @x-GHz</b>	22 @2-GHz (1.1% UI)	250.60 @0.111-GHz (0.83% UI)
<b>RMS jitter (ps) @x-GHz</b>	2.859 @2-GHz (0.14% UI)	73.90 @0.111-GHz (0.25% UI)
<b>Power (mW) @x-GHz</b>	3.31 @1-GHz	0.20049972 @0.111-GHz
<b>Chip Area (mm<sup>2</sup>)</b>	0.019	0.0360491

(UI : unit interval, one clock cycle time, **UI of [1]** : 2ns, **UI of our work** : 30ns / The jitter types above are period jitter)

由上表分析總結我們設計 FMDLL 之性能，首先，還原文獻架構時，因為我們所設計 DCDL 延遲較大，因此最後輸出頻率範圍 4 - 114 MHz 不如文獻 [1] 之輸出頻率

0.7 - 2.0 GHz 來的高。從抖動量 (jitter) 可以分析，由於我們輸出時脈訊號週期較大，所以抖動量相對的會比較大，但是參考文獻 [2] 算出抖動量相對輸入訊號週期後發現，整體抖動量與 UI 的比例在我們的設計與文獻中的設計是差不多的。最後可以觀察到因為我們所設計的最大輸出時脈頻率較小，因此平均功耗相對會比較小。

在我們走完整個設計流程後，由表中能得知我們目前雖然 FMDLL 電路的基本功能得以實現，但是輸出頻率仍有大的進步空間，因此我們未來優化的方向應該從調整延遲線 DCDL 的精細度與 fine tune 的能力為目標，在目標為增加輸出頻率的願景下，也期望抖動量能控制在更小的範圍。

## 4. Conclusion

此次專題主要以實現論文 [1] 以全數位分數比延遲鎖定迴路 FMDLL 為主題改良並設計出數位電路架構，利用全新的模式選擇 (Select Logic)、和諧波鎖定偵測器 (HLD)、相位偵測器 (PD) 與逐漸逼近暫存器 (SAR) 來達到快速鎖定與微小的時鐘偏差。此 SAR-based FMDLL 使用 TSMC 90nm CMOS 製程來設計，並設定乘頻常數規格  $N = \{1, 4, 5, 8, 10\}$  以及  $M = \{1, 2, 3\}$ ，在不斷測試輸入時脈訊號乘以不同  $N/M$  倍後得到輸出頻率範圍為 4.00 - 114.32 MHz，且此 FMDLL 能在  $4 \times k \times M$  ( $k$  為  $Q$  控制碼的位元數) 個週期內達到快速鎖定。

在此學期專題結束前，藉由重新組織 RTL 電路的 Verilog code 與優化內部時序特性，我們的目標會放在得到更高的輸出頻率與更精細的延遲調整結構，最後希望輸出頻率能一步一步接近 GHz 之規模。

最後，此 FMDLL 擁有低電壓以及低功耗的優點，且簡易的電路架構與在不同製程下的設計便利性使得晶片上產生時脈訊號的應用更受歡迎，而在製程愈來愈精細、電路愈來愈複雜的情況下，低時鐘偏差、大範圍操作頻率與頻率調節精細度也逐漸成為電路中不可疏漏的性能指標，在應用上可以說是越來越重要。

## 5. Reference

- [1] Jongsun Kim, and Sangwoo Han, "A Fast-Locking All-Digital Multiplying DLL for Fractional-Ratio Dynamic Frequency Scaling" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 65, NO. 3, MARCH 2018
- [2] Shi-Yu Huang, "Timing Circuit Design and Applications", Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, 2023.
- [3] Shi-Yu Huang, "Cell-based IC Design and Implementation", Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, 2023.

## 6. Plan Management and Teamwork

此次專題之主題是黃錫瑜教授希望我們以上學期修習「時序電路設計與應用」這門課的知識為基礎，將論文內容實現出來，我們實作的方式主要是以硬體語言 Verilog 進行全數位電路設計，透過電機系工作站和 TSRI 的 Eda Cloud 使用 CAD tool 如 Innovus、Laker、nWave 等工具進行模擬。因為此專題進行期間，每兩個星期會進行一次會議，因此平時組員有空就會一起討論這禮拜要完成什麼進度。在實作過程中也遇到了不少問題，其中最難解決的是一些電路中的時序問題，由於系統處在震盪模式，所以免不了產生毛刺，這會連帶影響其他模塊的邏輯判斷，如果沒有將此問題解決，不僅會有 Timing Violation 的問題，還會除出錯誤的頻率，因此在專題進行期間，前半部分時間在設計電路與確認其功能正確性，而後半部分的時間都在處理與優化一些時序上的問題。

至於分工的部分，我們平常會一起討論也會將彼此的進度上傳到 GitHub，在參考對方程式碼的同時也會加入自己創新的部分各自測試結果，由於我們都是第一次自己嘗試從 Paper 中實現，因此在過程中也遇到不少問題，我們彼此探討如何解決，如果有我們無法解決的問題才會請教實驗室學長姊，也感謝張書瑜學長與周郁欣學姊在我們專題進行期間熱心協助我們解決問題，也幫助我們申請 TSRI 資源供給我們使用。最後感謝黃錫瑜教授提供我們機會與實驗室資源，我們不只是在學術上更了解時序電路這塊領域，更多的是從教授平時的提醒中學會自發性解決問題、不斷嘗試再嘗試的精神和與同儕或前輩在處理任何事情上的態度與方法。最後，結合了有效的團隊合作和教授與學長姐的協助，在有限的時間內，嘗試各種電路的可能性，修改過無數個版本後才能得出最完美的結果。