第 5 章

訂單-查找和因式分解

逆量子傅立葉變換和量子傅立葉變換是實現傅立葉變換的量子電路，它們可以用來解決各種有趣的問題。在本章中，我們現在介紹其中兩個最有趣的問題，分別是求*序問題*和*因式分解問題*。米勒於 1976 年證明解決求序問題等價於解決因式分解問題。對於 RSA 公鑰密碼系統，人們目前已經安裝了超過 4 億份其演算法副本，它是用於互聯網和萬維網安全的主要密碼系統。 RSA 公鑰密碼系統的安全性取決於將大自然數分解為兩個大素數的問題在經典電腦上是棘手的。

Shor 的求序演算法可以比任何傳統計算機更快地解決求序和指數因式分解問題。利用 Shor 演算法將 1024 位元的大自然數分解為兩個質數的產生 每個 512 位，Imre 和 Ferenc 在 [Imre and Ferenc 2005] 中顯示執行時間約為 0.01 秒。這就是說，一旦其可靠的實體實現在市場上可用，Shor 的演算法將使 RSA 公鑰密碼系統過時。在本章中，我們先介紹一些數論背景。接下來，我們解釋尋序問題如何意味著分解能力。我們也解釋了 shor 演算法如何解決尋序問題。接下來，我們將描述如何編寫量子演算法來實作 Shor 演算法，以解決求序和因式分解問題中最簡單的情況。

5.1 數論基礎介紹

我們將*整數*集合 ***Z*** *表示*為 { …, –3, –2, –1, 0, 1, 2, 3, …}。我們常將*非*負*整數*集合稱為 {0, 1, 2, 3, }，將…*正整數*集合稱為 {1, 2, 3, …}。這就是說，0（零）是非負整數集合中的一個元素，而*不是正整數*集合中的一個元素。我們有時會說自然數表示*正整數*，自然數的集合是{1,2,3, …}。

更正式地說，給定任何正整數 w 和 n ，我們用以下形式唯一地表示 w

$$w = q \times n + r \quad , (5.1)$$

其中 q *是一個非負整數，它是* w 除以 n 的商數（結果）和*餘數* r *的*範圍是 0 到 $(n-1)$。如果餘數 r 的值等於 0，則我們說在這種情況下 n 是 w 的*因數*或*除數*。否則， n 不是 w *的因子*。請注意，1 和 w 始終是 w *的因子*。模算術只是普通算術，我們只關注餘數。我們使用符號 $(\text{mod } N)$ 來指出我們正在使用關於正整數 N *的模算術*。例如，由於 1、3、5、7、9 和 11 除以 2 時都具有相同的餘數 (1)，

因此我們寫為 1 = 3 = 5 = 7 = 9 = 11 (mod 2)。

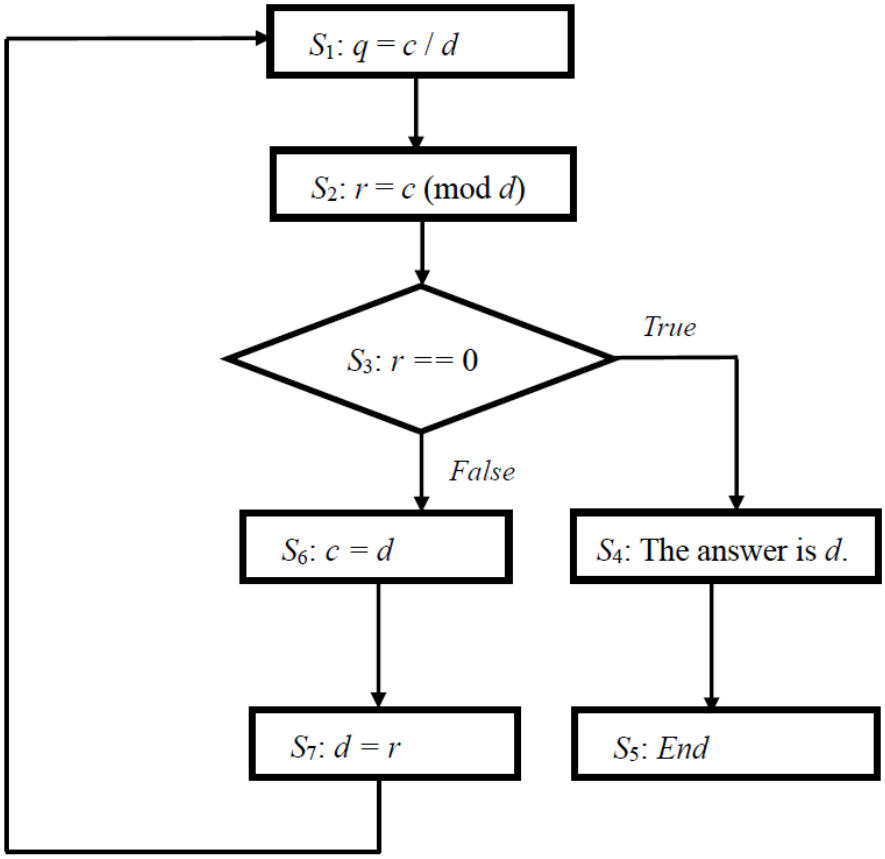
整數 c 和 *d* 的最大公約數或*因數是同時為 c* 和 *d* 的因數或因數的最大整數。我們將這個數字寫成 gcd(*c* , *d*)。例如，14 和 10 的最大公約數或因數是 2。然後選出兩個列表中等於二(2) 的最大公共元素。如果整數 *c* 和 *d* 的最大公約數為 1 ，則稱它們*互質*。*質數*是大於 1 的整數，只有它本身和 1 作為因數。如果一個數是大於 1 的整數且不是質數，那麼我們就稱它為合數。前幾個質數是 2, 3, 5, 7, 11, 13, 17, 19, 23, ….關於正整數最重要的一個事實也許是我們可以將它們唯一地表示為因數（質數）的乘積。令 *b* 為*任何*大於 1 的整數。

$$乙 = p_1^{b_1} \times p_2^{b_2} \times \ldots \times p_n^{b_n}， \quad （5.2）$$

其中$p_1, p_2, \ldots, p_n$是不同的質數， $b_1, b_2, \ldots, b_n$是正整數。對於小數，透過反覆試驗找到素因數分解是非常容易的。例如，對於一個小數 10，它的質因數分解為 $10 = 2^1 \times 5^1$.儘管為了尋找一種能夠有效地確定大數素因數分解的方法付出了巨大的努力，但在數位計算機中還沒有已知的有效方法來完成這項任務。

5.2 歐幾裡得演算法描述

*歐幾裡得演算法*是計算最大公約數的更有效的方法。圖 5.1 是歐幾裡得演算法的流程圖。我們用一個例子來解釋歐幾裡得演算法是如何計算出最大公約數的。此範例是確定兩個正整數 *c* = 15 和 *d* = *12* 的最大公約數。個執行語句 S 1 開始，得到商 q = *15* / 12 = *1* 。次執行圖 5.1 的語句 S_3 時，它會傳回*錯誤的*。因此，接下來，從第一次執行圖 5.1 中的語句 S_6 開始，它得到被除數 *c* = 12 的新值。值 *d* = 3。



演算法的流程圖。

接下來，從圖 5.1 的語句 S_1 第二次執行，得到商 $q = 12 / 3 = 4$ 。 . *由於* r 的值等於 0，因此在第二次執行圖 5.1 中的語句 S_3 時，它會傳回 *true* 。接下來，從第一次執行圖 5.1 的語句 S_4 開始，得到答案是 3 。 5 開始，答案是 3 。

歐幾裡得演算法 e 消耗哪些資源？我們假設兩個正整數 c 和 d 可以表示為每個最多 L 位的位元串。這表示商 q 和餘數 r *的長度都*不能超過 L 位。因此，我們可以假設使用 L 位元算術來完成每個計算。從圖 5.1 可以看出，除法和餘數運算是歐幾里德演算法的核心。最多使用 $O(L)$次的除法和餘數運算就可以完成歐幾里德演算法。由於每次除法和求餘運算都需要 $O(L^2)$ 次運算，因此歐幾里德演算法的總成本為 $O(L^3)$。

5.3 二次同餘的說明

我們假設 N *是*一個合數 n 位。 1 和 N *是* N *本身*的兩個微不足道的因子。我們也假設有一個函數β$\{X \mid 0 \le X \le N\} \rightarrow \{X^2 \pmod N\}$ 。函數的定義域β為$\{X \mid 0 \le X \le N\}$，其範圍為$\{X^2 \pmod N\}$。如果有一個整數 $0 \le X \le$ *使得* $\beta(X) = X^2 = C$

(mod N)，即同餘有解，則 C 據說是二次同餘 (mod ）。二次同餘 (mod N) 是 [Manders 和 阿德曼 1978]。若 C 的值等於 1，則 $X^2 = 1 \pmod N$ 的四個整數解分別為 b 、 $N - b$ 、1 和 $N - 1$，其中 $1 < b < (N / 2)$ 且 $(N / 2) < N - b < N - 1$. 1 和 $N - 1$ 是平凡解， b 和 $N - b$ 是非平凡的解決方案。這是二次同餘的特例 (mod N)，這仍然是一個 NP 完全問題。**引理 5-1** 用於表明，如果我們能找到一個非平凡的解 X ，*我們就可以確定因子* $N \neq \pm 1 \pmod N$ 代入方程式 $X^2 = 1 \pmod N$。

引理 5-1 ：我們假設 N *是* n 位合數， X *是方程式* $X^2 = 1 \pmod N$ 在 0 範圍內的≤非平凡解 $X \leq N$，即既不是 $X = 1 \pmod N$ 也不是 $X = N - 1 = -1 \pmod N$。然後至少 gcd(N , $X - 1$) 和 gcd(N , $X + 1$) *是* N 的重要因子，可以使用歐幾里德演算法透過 $O(n^3)$ 運算來確定。

證明：

因為 $X^2 = 1 \pmod N$，所以 N *一定*能除 $X^2 - 1 = (X + 1) \times (X - 1)$。自 $X \neq 1$ 和 $X \neq$ *氮* -1.必然是 N 不整除($X + 1$)且不整除($X - 1$)。這就是說， N 必須與 ($X + 1$) 和 ($X - 1$) 且 $1 < X < N - 1$. 因此，我們得到 $X - 1 < X + 1 < N$ 。由條件 $X - 1 < X + 1 < N$ ，我們知道公因數不可能是 N 本身。應用歐幾裡得演算法進行 $O(n^3)$ 次運算，我們可以計算出 gcd(N , $X - 1$) 和 gcd(N , $X + 1$) ，因此得到一個不平凡的因數 N 。■

我們考慮一個例子，其中 N 等於 15，並且任何給定函數β為{ X |0≤ X ≤15} →{ X^2(模 15)}。給定函數β的定義域是 { X |0≤ X ≤15}，其範圍為{ X^2(mod 15)}。(X)從第一個輸入零到最後一個輸入十五的十六個輸出β依序為 0、1、4、9、1、10、6、4、4、6、10、1、9、4、 1 和 ^{0} 。因此， $X^2 = 1 \pmod{15}$ 的四個整數解分別是 4、11、1 和 14。1 和 14 是平凡解。 4 和 11 是重要的解決方案。因為 $4^2 = 1 \pmod{15}$ 且 $11^2 = 1 \pmod{15}$，所以 15 一定能整除 $4^2 - 1 = (4 + 1) \times (4 - 1)$ 和 $11^2 - 1 = (11 + 1) \times (11 - 1)$。因此，15 必須與 (4 + 1) 和 (4 1) 之一具有公因數，且 15 必須與 (11 + 1) 和 (11 −1)之一具有公因數。−也就是說，使用歐幾裡得演算法，我們可以算出 gcd(15, 5) = 5 和 gcd(15, 3) = 3 或 gcd(15, 12) = 3 和 gcd(15, 10) = 5。這表示 15 的質因數分解為 $15 = 5 \times 3$。

5.4 連分數簡介

實數和整數的連續體之間存在著許多有價值的連結。*連分數*理論就是這樣一種美妙的連結。如果 c 和 d 是整數，那麼我們稱(c / d)為*有理分數*或*有理數*。*有*

*限簡單連分數*由正整數的有限集合 q [1], q [2], q [3], …, q [i]表示，

$$(q[1], q[2], q[3], \ldots, q[i]) = q[1] + \frac{1}{q[2] + \frac{1}{q[3] + \cdots + \frac{1}{q[i]}}}. \quad (5.3)$$

我們表示第 k *個*收斂（$1 \le k \le i$）將該連分數變成

$$(q[1], q[2], q[3], \ldots, q[k]) = q[1] + \frac{1}{q[2] + \frac{1}{q[3] + \cdots + \frac{1}{q[k]}}}. \quad (5.4)$$

圖 5.2 是連分式演算法的流程圖。 We can use the continued fraction al algorithm to determine a finite collection q [1], q [2], q [3], …, q [i] of positive integers in (5.3) for representing continued fraction of a *rational fraction* , (*光碟*) .因此，應用右側等價 在(5.3)中我們可以將 c / d *描述*為

$$\frac{c}{d} = q[1] + \frac{1}{q[2] + \frac{1}{q[3] + \cdots + \frac{1}{q[i]}}}。 \quad (5.5)$$

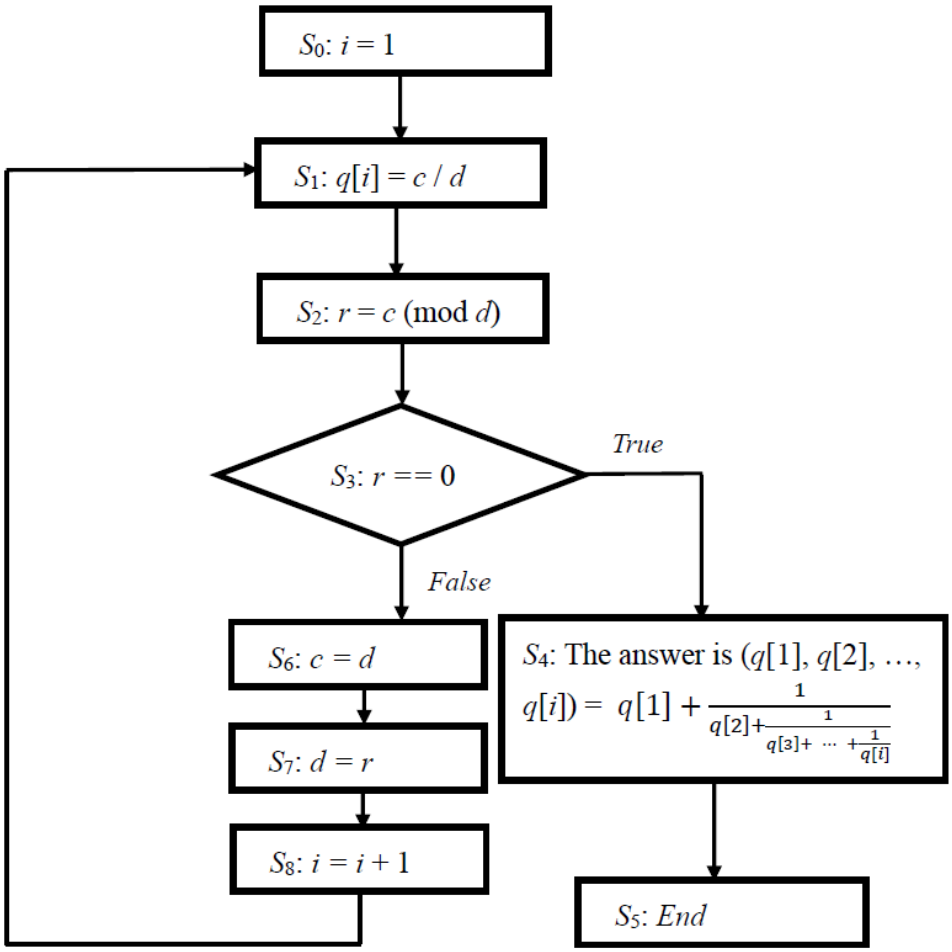


圖 5.2 ：連分數 a 演算法的流程圖。

我們用一個例子來解釋圖 5.2 中的連分式演算法是如何實現的 如果 $c = 31$ 且 $d = 13$ 以及相應的收斂，則確定 (c / d)的連續分數表示。從第一次執行語句 S_0 到語句 S_2 ，得到 $i = 1$、 $q[1] = c / d = 31 / 13 = 2$ 和 $r = 31 / (\text{mod } 13) = 5$。13) 化為整數和小數部分，並反轉其小數部分，

$$\frac{31}{13} = 2 + \frac{5}{13} = 2 + \frac{1}{\frac{13}{5}}。 \qquad (5.6)$$

r 的值等於 5，所以從第一次執行語句 S_3 開始，它回傳 *false* 。因此，接下來，從第一次執行語句 S_6 到語句 S_8 ，得到*分子的新值* c 等於 13，*分母的新值* d 等於 5，索引變數 i *的值*等於 2。

S_1 到語句 S_2 的第二次執行，得到 $q[2] = c / d = 13 / 5 = 2$ 和 $r = 13 / (\text{mod } 5) = 3$ –。反轉–習慣 (13 / 5), 給予

$$\frac{31}{13} = 2 + \frac{1}{2+\frac{3}{5}} = 2 + \frac{1}{2+\frac{1}{\frac{5}{3}}}。 \quad (5.7)$$

r 的值等於 3，因此從第二次執行語句 $S_{3\text{開始}}$，它會傳回 *false* 。因此，接下來，從第二次執行語句 S_6 到語句 S_8 ，得到分子的新值 *c* 等於 5，分母 *d 的新值*等於 3，索引變數 *i 的值*等於 3。

接下來，從第三次執行語句 S_1 到語句 S_2 ，它得到 *q [3*] = *c* / *d* = 5 / 3 = 1 和 *r* = 5 / (mod 3) = 2 –。反轉–習慣 (5 / 3), 給予

$$\frac{31}{13} = 2 + \frac{1}{2+\frac{1}{1+\frac{2}{3}}} = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{\frac{3}{2}}}}。 \quad (5.8)$$

r 的值等於 2，因此從第三次執行語句 $S_{3\text{開始}}$，它會傳回 *false* 。因此，接下來，從第三次執行語句 S_6 到語句 S_8 ，得到分子的新值 *c* 等於 3，分母 *d 的新值*等於 2，索引變數 *i 的值*等於 4。

接下來，從語句 S_1 到語句 S_2 的第四次執行，它取得 *q* [4] = *c* / *d* = 3 / 2 = 1 和 *r* = 3 / (mod 2) = 1。–然後反轉–習慣 (3 / 2), 給予

$$\frac{31}{13} = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{1+\frac{1}{2}}}} = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{1+\frac{1}{\frac{2}{1}}}}}。 \quad (5.9)$$

r 的值等於 1，因此從第四次執行語句 $S_{3\text{開始}}$，它會傳回 *false* 。這樣，接下來從第四次執行語句 S_6 到語句 S_8 ，就得到分子的新值 *c* 等於 2，分母 *d 的新值*等於 1，索引變數 *i 的值*等於 5。

接下來，從第五次執行語句 S_1 到語句 S_2 ，得到 *q* [5] = *c* / *d* = *2* / 1 = 2 和 *r* = 2 / (mod 1) = 0 。 ，這是將(2 / 1) 分成整數部分和小數部分，而*不是*反轉其小數部分。這意味著 (2 / 1) $\frac{0}{1}$ = 2 + = 2 –。–

$$\frac{31}{13} = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{1+\frac{1}{2+\frac{0}{1}}}}} = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{1+\frac{1}{2}}}}。 \quad (5.10)$$

r 的值等於 0，所以從第五次執行語句 $S_{3\text{開始}}$，它回傳 *true* 。因此，接下來，從第一次執行語句 $S_{4\text{開始}}$，答案就是 (31 / 13) 的連分數表示

$$\frac{31}{13} = (q[1] = 2, q[2] = 2, q[3] = 1, q[4] = 1, q[5] = 2) = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{1+\frac{1}{2}}}}. \quad (5.11)$$

接下來，從第一次執行語句 $S_{5\text{開始}}$，終止連分數 a 演算法的執行。對於有理數 (31 / 13)，第一個收斂到第五個收斂依序為 $(q[1]) = 2$, $(q[1], q[2]) = 2 + \frac{1}{2} = \frac{5}{2}$, $(q[1], q[2], q[3]) = 2 + \frac{1}{2+\frac{1}{1}} = \frac{7}{3}$, $(q[1], q[2], q[3], q[4]) = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{1}}} = \frac{12}{5}$且 $(q[1], q[2], q[3], q[4], q[5]) = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{1+\frac{1}{2}}}} = \frac{31}{13}$。

圖 5.2 中的連分數演算法消耗 e 哪些資源來獲得有理數 $c / d > 1$ 的連分數展開，其中 c 和 d 是 L *位元整數*? 這就是說， 1 的 $q[k]$ 有多少個值 $\le k \le$ (5.3) 中的 i *必須由*圖 5.2 中的連分數 al 演算法決定。從圖 5.2 的連分數演算法中的語句 S_1 和語句 $S_{2\text{中，每個商}} q[k]$ 為 $1 \le k \le i$ 至多 L 位長，餘數 r 至多 L 位長。因此，我們可以假設使用 L 位元算術來完成每個計算。從圖 5.2 可以看出，除法和餘數運算是連分數演算法的核心。最多應用 $O(L)$ *次*除法和餘數運算即可完成連分數演算法。由於每次除法和餘數運算都需要 $O(L^2)$次運算，因此連分數演算法的總成本為 $O(L^3)$。

5.5 尋找訂單和保理

我們假設 N *是正整數*， X 和 N 的最大公約數是 1 為 $1 \le X \le N$ 和 X *與* N *互質*。 X 模 N 的*階數*是最小*正*整數 r ，使得 $X^r = 1 \pmod N$。排序-*查找 問題是在*給定 X 和 N *的情況*下計算 r 。經典計算機上沒有有效的演算法來解決求序問題和因式分解問題。然而，解數因式分解問題就等於解決求序問題。這就是說，如果有一種有效的演算法來解決量子電腦上的求序問題，那麼它就可以快速解決數位因式分解問題。我們使用**引理 5-2 來證明**，如果我們能夠找到 X 模 N 的*階* r 以滿足 $X^r = 1 \pmod N$ 並且甚至使得一個非平凡解，$X^{\frac{r}{2}}$我們就可以確定 N *的因子* $\neq \pm 1 \pmod N$ 代入方程式 $X^r = (X^{\frac{r}{2}})^2 = 1$（模 N ）。同時，我們使用**引理 5-**

3 來展示兩個整數 c 和 d 的最大公約數的表示定理。

引理 5- 2 ：我們假設 N *是* n *位*合數，且 X 模 N 的*階數* r 滿足 $X^r = 1 \pmod N$ 且甚至非平凡解 $X^{\frac{r}{2}} \neq \pm 1 \pmod N$ 代入方程式 $X^r = (X^{\frac{r}{2}})^2 = 1 \pmod N$ 在 0 範圍內 $\leq X^{\frac{r}{2}} \leq$ *尼*。那就是既不 $X^{\frac{r}{2}} = 1 \pmod N$ 也不 $X^{\frac{r}{2}} = N - 1 = -1 \pmod N$。然後至少 $\gcd(N, X^{\frac{r}{2}} - 1)$ 和 $\gcd(N, +1)$ 是 $X^{\frac{r}{2}} N$ 的重要因子，可以使用歐幾里德演算法透過 $O(n^3)$ 運算來確定。

證明：

由於 $X^r = (X^{\frac{r}{2}})^2 = 1 \pmod N$ ，一定是 N 整除 $(X^{\frac{r}{2}})^2 - 1 = (X^{\frac{r}{2}} + 1) \times (X^{\frac{r}{2}} - 1)$。因為 $X^{\frac{r}{2}} \neq 1$ 和 $X^{\frac{r}{2}} \neq$ *氮* -1.必然是 N 既不整除 $(X^{\frac{r}{2}} + 1)$ 又不整除 $(X^{\frac{r}{2}} - 1)$。這意味著 N 必須與 $(X^{\frac{r}{2}} + 1)$ 和 $(X^{\frac{r}{2}} - 1)$ 且 $1 < X^{\frac{r}{2}} < N - 1$. 因此，我們得到 $X^{\frac{r}{2}} - 1 < X^{\frac{r}{2}} + 1 < N$ 。從條件來看 $X^{\frac{r}{2}} - 1 < X^{\frac{r}{2}} + 1 < N$ ，我們看到公因子不可能是 N 本身。使用歐幾裡得演算法進行 $O(n^3)$ 次運算，我們可以計算 $\gcd(N, X^{\frac{r}{2}} - 1)$ 和 $\gcd(N, +1)$ ，從而得到 $X^{\frac{r}{2}} N$ 的重要因數。■

引理 5- 3 ：兩個整數 c 和 d 的最大公約數是可以寫成 c 形式的最小正整數 $\times$ *你* $+ d \times v$ ，其中 u 和 v 是整數。

證明：

令 $t = c \times$ *你* $+ d \times v$ 是寫成這種形式的最小正整數。令 w *是* c 和 d 的最大公約數。因此， w *是* c 和 d 的約數，並且是 t *的約數*。這意味著 $w \leq t$ 。為了完成證明，我們展示 $t \leq w$ 藉由證明 t *是* c 和 d 的約數。證明是透過反證法。我們假設 t 不是 c *的約數*。那麼 $c = q \times t + r$ ，其中餘數 r *在 1 到* t *的*範圍內 -1. 重新整理這個方程式 $c = q \times t + r$ 並使用 $t = c \times$ *你* $+ d \times v$ ，我們得到 $r = c \times (1 - q \times) + d \times (-q \times v)$ 是正整數，是 c 和 d *的線性組合*。因為 r 小於 t ， **這與** t 的定義相矛盾，t 是用 c 和 d 的線性組合所寫的最小正整數。因此，我們推論 t 必須整

除 c 。類似地，根據對稱性，t 也必須是 d 的約數。這意味著 $t \le$ 華和瓦 $\le t$ 。因此，我們完成證明。■

引理 5-4 ：我們假設整數 a 整除整數 c 和整數 d 。然後 a 除 c 和 d 的最大公約數。

證明：

由**引理 5-3 可知**， c 和 d 的最大公約數是 $c \times$ 你$+ d \times v$ ，其中 u 和 v 是整數。因為 a 既能整除 c 又能整除 d ，因此它也必須能整除 $c \times$ 你$+ d \times v$.因此，我們立即推論 a 整除 c 和 d 的最大公約數。■

在模算術中，數字 c 何時具有乘法逆元？這是問，給定 c 和 N ，何時存在 d 使得 $c \times d = 1 \pmod N$)？我們考慮一個例子，其中 $3 \times 4 = 1 \pmod{11}$。這顯示數字 3 在模 11 的算術中具有乘法逆元 4 。質數：如果整數 c 和整數 d 的最大公約數為 1（一），則它們是互素數。例如，3 和 11 是互質的，因為 3 的正因數是 1 和 3，11 的正因數是 1 和 11 。 。同時，我們利用**引理 5-6** 證明 X 模 N 的階數 r 滿足 $r \le$ 尼。

引理 *5-5* ：設 N 為大於 1 的整數。

證明：

我們使用 gcd(X , N) 來表示 X 和 N 的最大公約數。我們假設 X 具有乘法逆元，我們定義 X^{-1} 模 N 。然後 $X \times X^{-1} = 1 \pmod N$)。這給了 $X \times X^{-1} = u \times N + 1$ 對於某個整數 u ，因此 $X \times X^{-1} + (-u) \times N = 1$ 。 $\times X^{-1} + (-u) \times N = 1$ 。

相反，如果 gcd(X , N) = 1，則根據**引理 5-3** 必須存在整數 y 和整數 z 使得 $X \times y + z \times N = 1$. 兩邊取模運算後得到 $X \times y \pmod N + z \times N \pmod N = 1 \pmod N$)。作為 $z \times N \pmod N = 0$, 那我們有 $X \times y \pmod N = 1 \pmod N$)。這意味著 X 的餘數 $\times y$ 模 N 等於 1。因此，我們得到 $X \times y = 1 \pmod N$)。因此， $y = X^{-1}$ 是 X 的乘法逆元。■

引理 5-6 ： 設 N 為大於 1 和 1 的整數≤ $X \le$ 尼。 X 和 N 互質， r 是最小正整數，使得 $X^r = 1 \pmod N$)。然後 $r \le$ 尼。

證明：

X的不同階值序列： $X^0 \pmod N$ 、 $X^1 \pmod N$、 $X^2 \pmod N$， …, X^{N-1} (模N), X^N （模N ）。模運算下只能有N *個*唯一值 0, 1, 2, …, N –上述序列中的$X^i \pmod N$為 1 。如果上述序列中的項數多於N ，則在應用模運算時，某些$X^i \pmod N$ *將具有相同的值（*上述序列中有$N + 1$ *項）*。例如 ，設$N = 5$且$X = 2$ 。

因此，在上述序列的不同項中，有兩項在模運算下是等價的， $X^n = X^m \pmod N$，其中我們可以不失一般性地假設$n > m$且n ，*米* ≤ *尼*。由於根據**引理 5-5** $\gcd(X, N) = 1$，我們知道存在X的乘法逆元X^{-1}使得$X \times X^{-1} = 1 \pmod N$。因為$X$和$N$的最大公約數是1，所以$X^m$和$N$的最大公約數等於1。由**引理 5-5** ，$\gcd(X^m, N) = 1$，我們知道存在$X^m$的乘法逆$X^{-m}$使得$X^m \times X^{-m} = 1 \pmod N$。接下來，將模運算兩邊$X^n = X^m \pmod N$ 乘以X^{-m}以獲得$X^n \times X^{-米} = X^{米} \times X^{-m} \pmod N$且$X^{nm-} = X^{mm-} \pmod N = X^0 \pmod N = 1 \pmod N$ 。從上面的陳述我們有$r = n - 米$。此外，當$n, m \le N$且$n > m$ ，因此$r = n - 米 \le 尼$。■

5. 6 計算 2 模 15 的階數和 15 的質因數分解

我們**想要**找出 N = 15 的**質***因數*分解。 = 2 和 N = *15* 和為 1（一）。這就是說， **X** = 2 與 N = 15 *互*質。 命令 r 對 2 模 15 滿足 r 15. 由於表示≤N = *15* 的位數為 4 位長，我們也只需要使用 4 位來表示 r 的值。

確定 命令 r 對 2 模 15 相當於確定週期 r 給定的神諭函數 P_f: { $r_1 r_2$ *右*$_3$ *右*$_4$ | ∀ *研發* ∈{0, 1} 為 $1 \le d \le 4$} →{$2^{r_1 r_2 r_3 r_4} \pmod{15}$ |∀ *研發* ∈{0, 1} 為 $1 \le d \le 4$}。 P_f的周期 r 滿足 $P_f(r_1 r_2 r_3 r_4) = P_f(r_1 r_2 r_3 r_4 + r)$ 到任兩個輸入 ($r_1 r_2 r_3 r_4$) 和 ($r_1 r_2 r_3 r_4 + r$)。 P_f的 16 個輸出，每個輸入來自 $r_1{}^0 r_2{}^0 r_3{}^0 r_4{}^0$ 至 $r_1{}^1 r_2{}^1 r_3{}^1 r_4{}^1$ 依序為 1 , 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, *4* 和8 。輸出的周期數等於四。 P_f的周期 r 是 P_f的頻率 f *的倒數*。因此，我們得到 $r = \frac{1}{\frac{f}{16}} = \frac{16}{f} = \frac{16}{4} = 4$ 和 $r \times f = 4 \times 4 = 16$。

另一方面，我們將 P_f *的輸入域*視為時域，輸出視為訊號。計算 命令 r 對 2 模 15 相當於確定了時域（輸入域）訊號的周期 r 和頻率 f 。因為每個輸入的輸出都來自 $r_1{}^0 r_2{}^0 r_3{}^0 r_4{}^0$ 至 $r_1{}^1 r_2{}^1 r_3{}^1 r_4{}^1$ 依序為 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4 和 8，我們將這十六個輸入值作為對應的十六個時間單位和十六個輸出作為十六個訊號樣本。每個樣本對 P_f的輸出進行編碼。 *輸出可以*取 1、2、4 *或*8 。 $r_3{}^0 r_4{}^0$ 至 $r_1{}^1 r_2{}^1 r_3{}^1 r_4{}^1$ 對應於從零到十五的十六個時間單位。

我們用圖 5.3 來解釋為什麼要計算 命令 r 對 2 模 15 相當於確定了時域（輸

入域）訊號的周期 r 和頻率 f 。在圖 5.3 中，橫軸代表時域，其中包含 $P_{f\text{的輸入域}}$，縱軸代表由 $P_{f\text{的16個輸出組成的訊號}}$。為了表達方便，我們用變數 k 來表示每個二進位輸入的十進位值，並用 $2^k \bmod 15$ 來表示$2^{r_1 r_2 r_3 r_4}$ （模 15）。

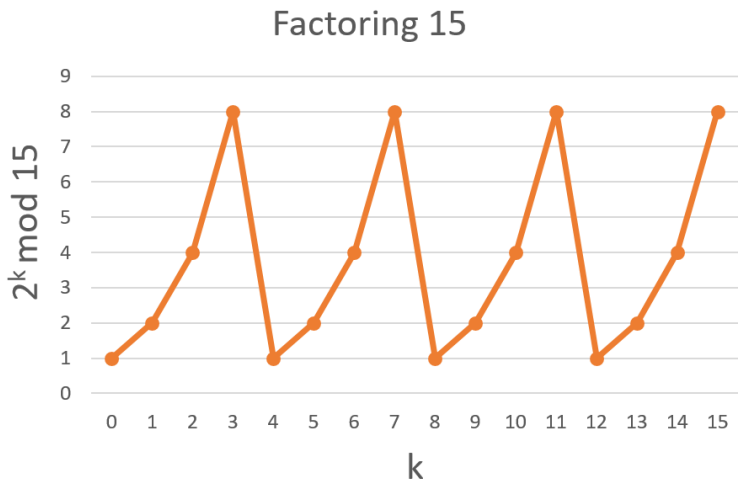


圖 5.3 ：從給定神諭函數 P_f的 16 個輸出中取樣 16 個點：｛$r_1 r_2$ *右* $_3$ *右* $_4$ | ∀ *研發* ∈{0, 1} 為 $1 \le d \le 4$} →{$2^{r_1 r_2 r_3 r_4}$ (模 15) |∀ *研發* ∈{0, 1} 為 $1 \le d \le 4$}。

給定神諭函數 P_f 中儲存的隱藏模式和訊息是訊號*四次旋轉回其第一個*訊號（輸出為 1）。其訊號*四次*循環回到其*第二個*訊號（輸出為 2）。其訊號循環回到其*第三*訊號（輸出為 4）*四次*，其訊號循環回到其*第四*訊號（輸出為 8）*四次*。這表示每十六個時間單位有四個訊號週期，且訊號的頻率 f 等於四。

因為圖 5.3 訊號的周期 r 是訊號頻率 f *的倒數*，所以訊號的$\frac{1}{\frac{4}{16}}$週期 $r = 16 / 4 = 4$ 。 圖 5.3 滿足 $P_f(r_1{}^0 r_2{}^1 r_3{}^0 r_4{}^0) = 2^{r_1^0 r_2^1 r_3^0 r_4^0} (\text{mod } 15) = 2^4 (\text{mod } 15) = 16 (\text{mod } 15) = 1 (\text{mod } 15)$ ，因此 $P_{f\text{中訊號的}}$週期 $r = 4$ 相當於階數 $r = 4$ of $2 \text{ modulo } 15$ 。$2^{r_1 r_2 r_3 r_4}$ （模 15）。由於 $r = 4$ 是偶數且小於 15，根據**引理 5-2** ，我們使用歐幾里德演算法來計算 $\gcd(15, 2^{\frac{4}{2}} + 1)$ 和 $\gcd(15, 2^{\frac{4}{2}} - 1)$。也就是說， $N = 15$ 的兩個非平凡因子分別是 *5* 和×3 。

因為 $(\frac{1}{\frac{4}{16}} = 16/4)$ 是一個有理數且是整數，所以我們使用圖 5.2 中的連分數 al 演算法 確定 (c / d) 的連分式表示（如果 $c = 16$ 且 $d = 4$）以及相應的收斂性，以解釋連分式 al 演算法在實際應用中的工作原理。從第一次執行語句 S_0 到語句 S_2 ，得到 $i = 1$、 $q[1] = c / d = 16 / 4 = 4$ 和 $r = 16 (\text{mod } 4) = 0$。 ）轉換其

整數和小數部分且不反轉其小數部分，

$$\frac{16}{4} = 4 + \frac{0}{4} = 4 \text{ 。} \qquad (5.12)$$

r 的值等於 0，因此從第一次執行語句 $S_{3\,開始}$，它會傳回 *true* 。因此，接下來，從第一次執行語句 $S_{4\,開始}$，答案是 (16 / 4) 的連分數表示

$$\frac{16}{4} = (q[1] = 4) = 4. \quad (5.13)$$

接下來，從第一次執行語句 $S_{5\,開始}$，終止連分數 a 演算法的執行。對於有理數 (16 / 4)，第一個收斂是 $(q[1]) = 4 = \frac{4}{1}$，它最接近($\frac{1}{\frac{4}{16}} = \frac{16}{4}$)並且實際上等於$\frac{16}{4}$。

這意味著第一個收斂($q[1]) = 4 = \frac{4}{1}$等於週期 $r = \frac{r}{1}$。因此，我們得到週期 r 等於分子 4 的第一個收斂。因為分子 $r = 4$ 第一個收斂的值小於 $N = 15$ ，分子 $r = 4$ 等價於 命令 $r = 2$ 的 4 模 15 滿足 $2^4 = 1 \pmod{15}$。

5.7 求 2 模 21 的階數和 21 的質因數分解

我們**想要**搜尋 $N = 21$ 的*質因數分解*。 $= 2$ 且 $N = 21$ 為 1（一）。這顯示 $X = 2$ 與 $N = 21$ *互質*。 命令 r 的 2 模 21 滿足 r 21. 表示≤$N = 21$ *的*位數有 5 位長，因此我們只需要使用 5 位來編碼 r *的值*。

計算 命令 r 2 模 21 相當於計算出週期 r 給定的神諭函數 A_f: { $r_1 r_2$ *右* $_3$ *右* $_{4\,5}$ *號* | ∀ *研發* ∈{0, 1} 為 $1 \le d \le 5$} →{$2^{r_1 r_2 r_3 r_4 r_5}$ （模 21）|∀ *研發* ∈{0, 1} 為 $1 \le d \le 5$}。 A_f的周期 r 滿足 $A_f(r_1 r_2 r_3 r_4 r_5) = A_f(r_1 r_2$ *右* $_3$ *右* $_4 r_5 + r)$ 到任兩個輸入 ($r_1 r_2$ *右* $_3$ *右* $_4 r_5$) 和 ($r_1 r_2$ *右* $_3$ *右* $_4 r_5 + r$)。 A_f的 32 個輸出，每個輸入來自 $r_1{}^0 r_2{}^0 3_0 4^0 r_5{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 r_5{}^1$ 隨後為 1, 2, 4, 8, 16, 11, 1, 2, 4, 8, 16, 11, 1, 2, 4, 8, 16, 11, 1, 2, 4, 8, 16 、 11、1、2、4、8、16、11、1 和 *2* 。 A_f的周期 r 是 A_f的頻率 f *的倒數*。因此，我們得到 $r = \frac{r}{1} = \frac{1}{\frac{f}{32}} = \frac{32}{f}$和 $r \times f = 32 \times 1 = 32$。

另一方面，我們將 $A_{f\,的輸入域}$視為時域，輸出視為訊號。弄清楚 命令 2 模 21 的 r 相當於計算時域（輸入域）訊號的週期 r 和頻率 f 。 $r_1{}^0 r_2{}^0$每個輸入的輸

出 $3_0 4^0 r_5{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 r_5{}^1$ 隨後為 1, 2, 4, 8, 16, 11, 1, 2, 4, 8, 16, 11, 1, 2, 4, 8, 16, 11, 1, 2, 4, 8, 16 、 11、 1、 2、 4、 8、 16、 11、 1 和 2。 因此，我們將 32 個輸入值作為對應的 32 個時間單位，將 32 個輸出作為 32 個樣本訊號。每個樣本對 A_f 的輸出進行編碼。輸出可以取 1、2、4、8、16 或 11 。 $3_0 4^0 r_5{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 r_5{}^1$ 對應於從零到三十一的三十二個時間單位。

我們應用圖 5.4 來解釋為什麼確定 命令 2 模 21 的 r 相當於決定時域（輸入域）訊號的週期 r 和頻率 f 。在圖 5.4 中，橫軸表示包含 A_f 輸入域的時域，縱軸表示包含 A_f 的 32 個輸出的訊號。為了表達方便，我們用變數 k 來表示每個二進位輸入的十進位值，並用 $2^k \bmod 21$ 來表示$2^{r_1 r_2 r_3 r_4 r_5}$ （模 21）。

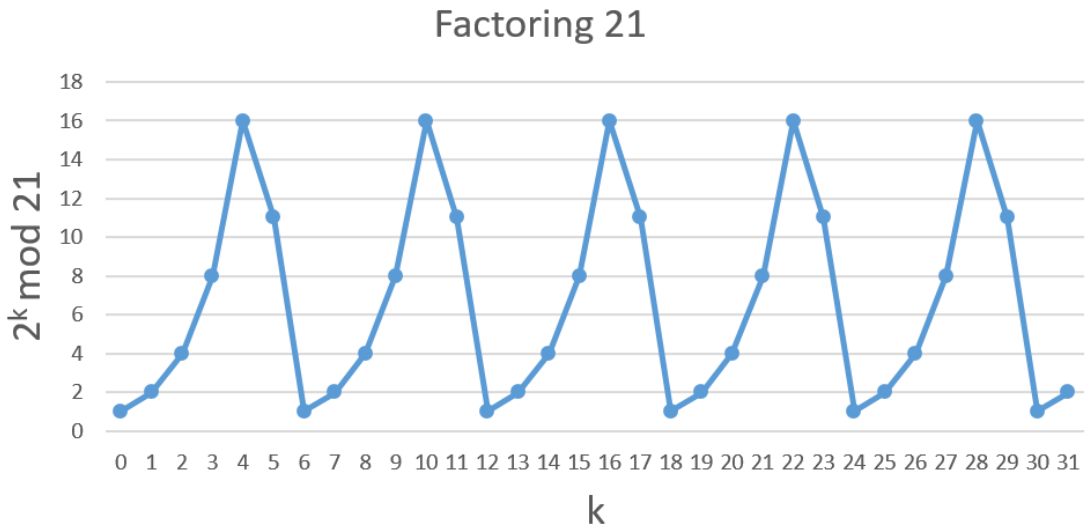


圖 5 . 4 ：從給定神諭函數 A_f 的 32 個輸出中取樣 32 個點： $\{r_1 r_2$ 右 $_3$ 右 $_4{}_5$ 號 $\mid \forall$ 研發 $\in\{0, 1\}$ 為 $1 \leq d \leq 5\} \rightarrow \{2^{r_1 r_2 r_3 r_4 r_5}$ （模 21） $\mid \forall$ 研發 $\in\{0, 1\}$ 為 $1 \leq d \leq 5\}$。

給定神諭函數 A_f 中儲存的隱藏模式和訊息是訊號旋轉回其第一個訊號（輸出為 1）六次。其訊號循環回到其第二個訊號（輸出為 2）六次。其訊號循環回到其第三個訊號（輸出為 4）五次，其訊號循環回其第四訊號（輸出為 8）五次。其訊號循環回到其第五訊號（輸出 16）五次，其訊號循環回其第六訊號（輸出 11）五次。也就是說每三十二個時間單位有($5\frac{2}{6}$)個訊號週期，訊號的頻率 f 等於 ($5\frac{2}{6}$) 。

由於圖 5.4 中訊號的周期 r 是訊號頻率 f 的倒數，因此訊號的$\frac{1}{\frac{5\frac{2}{6}}{32}}$周期 r = =

$\frac{1}{\frac{32}{\frac{6}{32}}}$= 6 / 1 = 6 $\frac{32}{\frac{32}{6}}$。 圖 5.4 滿足 $A_f(r_1{}^0 r_2{}^0 r_3{}^1 r_4{}^1 r_5{}^0) = 2^{r_1^0 r_2^0 r_3^1 r_4^1 r_5^0} (\text{mod } 21) =$ $2^6 (\text{mod } 21) = 64 (\text{mod } 21) = 1 (\text{mod } 21)$ ，因此 A_f 中訊號的周期 $r = 6$ 等於 2 模 21 的階 $r = 6$ 。$2^{r_1 r_2 r_3 r_4 r_5}$ （模 21）。因為 $r = 6$ 是偶數且小於 21，所以根據**引理 5-2** ，我們使用歐幾里德演算法來計算 gcd(21, $2^{\frac{6}{2}}+1$) 和 gcd(21,$2^{\frac{6}{2}}-1$)。這意味著 $N = 21$ 的兩個重要因子分別為 *3* 和×7 。

因為 ($\frac{1}{\frac{5^{\frac{2}{6}}}{32}}$= $\frac{1}{\frac{32}{\frac{6}{32}}}$= $\frac{32}{\frac{32}{6}}$= 6 / 1) 是一個有理數，且是整數，所以我們應用圖 5.2 中的連分數 al 演算法 確定 (c / d) 的連分式表示（如果 $c = 6$ 且 $d = 1$）以及相應的收斂性，以解釋連分式 al 演算法在實際應用中的工作原理。從第一次執行語句 S_0 到語句 S_2 ，得到 $i = 1$ 、 $q[1] = c / d = 6 / 1 = 6$ 和 $r = 6 (\text{mod } 1) = 0$。) 轉換其整數和小數部分且不反轉其小數部分，

$$\frac{6}{1} = 6 + \frac{0}{1} = 6 \text{ 。} \qquad (5.14)$$

r 的值等於 0，因此從第一次執行語句 $S_{3\text{開始}}$，它會傳回 *true* 。因此，接下來，從第一次執行語句 $S_{4\text{開始}}$，答案是 (6 / 1) 的連分數表示

$$\frac{6}{1} = (q[1] = 6) = 6. \quad (5.15)$$

接下來，從第一次執行語句 $S_{5\text{開始}}$，終止連分數 a 演算法的執行。對於有理數 (6 / 1)，第一個收斂是 (q [1]) = 6 = ，$\frac{6}{1}$它最接近($\frac{1}{\frac{5^{\frac{2}{6}}}{32}}$= $\frac{1}{\frac{32}{\frac{6}{32}}}$= $\frac{32}{\frac{32}{6}}$= $\frac{6}{1}$)並且實際上等於$\frac{6}{1}$。這顯示第一個收斂(q [1]) = 6 =$\frac{6}{1}$等於週期 $r = \frac{r}{1}$。因此，我們得到週期 r 等於分子 6 的第一個收斂。由於分子 r = 6 第一個收斂的值小於 N= 21 ，分子 r = 6 等價於 命令 $r = 2$ 的 6 模 21 滿足 $2^6 = 1 (\text{mod } 21)$。

5.8 計算 2 模 35 的階數和 35 的質因數分解

我們想要找出 $N = 35$ 的質因數分解。 $= 2$ 和 $N = 35$ 和為 1（一）。這意味著 $X = 2$ 與 $N = 35$ 互質。 命令 r 對 2 模 35 滿足 r 35. 表示≤$N = 35$ 的位數有 6 位長，我們只需要使用 6 位來編碼 r 的值。

計算 命令 r 2 模 35 相當於計算週期 r 給定的神諭函數 B_f: { $r_1 r_2$ 右 $_3$ 右 $_4$ $_5$ 號 r_6 | ∀ 研發 ∈{0, 1} 為 $1 \le d \le 6\} \rightarrow \{ 2^{r_1 r_2 r_3 r_4 r_5 r_6}$（模 35）|∀ 研發 ∈{0, 1} 為 $1 \le d \le 6\}$。 B_f的周期 r 滿足 $B_f(r_1 r_2 r_3 r_4 r_5 r_6) = B_f(r_1 r_2$ 右 $_3$ 右 $_4$ $_5$ 號 $r_6 + r)$ 到任兩個輸入 $(r_1 r_2$ 右 $_3$ 右 $_4$ $_5$ 號 $r_6)$ 和 $(r_1 r_2$ 右 $_3$ 右 $_4$ $_5$ 號 $r_6 + r)$。 B_f的前 24 個輸出，每個輸入來自 $r_1{}^0 r_2{}^0 3_0 4^0 5_0 r_6{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 5_1 r_6{}^1$ 隨後為 1, 2, 4, 8, 16, 32, 29, 23, 11, 22, 9, 18, 1, 2, 4, 8, 16, 32, 29, 23, 11, 22, 9 18、 B_f的中間二十四個輸出分別為 1、2、4、8、16、32、29、23、11、22、9、18、1、2、4、8、16、32、 29 , 23 , 11 , 22 , 9 和 18 。 、 B_f的頻率 f 等於每 64 個輸出的週期數。 B_f的周期 r 是 B_f的頻率 f 的倒數。因此，我們得到 $r = \frac{r}{1} = \frac{1}{\frac{f}{64}} = \frac{64}{f}$ 和 $r \times f = 64 \times 1 = 64$。

另一方面，我們將 B_f 的輸入域視為時域，輸出視為訊號。確定 命令 r 2 模 35 相當於計算出時域（輸入域）訊號的周期 r 和頻率 f 。每個輸入的前二十四個輸出來自 $r_1{}^0 r_2{}^0 3_0 4^0 5_0 r_6{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 5_1 r_6{}^1$ 隨後為 1, 2, 4, 8, 16, 32, 29, 23, 11, 22, 9, 18, 1, 2, 4, 8, 16, 32, 29, 23, 11, 22, 9 18、中間二十四個輸出分別為 1、2、4、8、16、32、29、23、11、22、9、18、1、2、4、8、16、32、29、23 、 11、22、 9 和 18 。 64 個輸入值作為對應的 64 個時間單位，將 64 個輸出作為 64 個訊號樣本。每個樣本對 B_f的輸出進行編碼。輸出可以取 1、2、4、8、16、32、29、23、11、22、9 或 18 。 $3_0 4^0 5_0 r_6{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 5_1 r_6{}^1$ 對應於從零到六十三的六十四個時間單位。

我們應用圖 5.5 來說明為什麼計算 命令 r 對 2 模 35 相當於確定了時域（輸入域）訊號的周期 r 和頻率 f 。在圖 5.5 中，橫軸表示由 B_f 的輸入域組成的時域，縱軸表示包含 B_f 的 64 個輸出的訊號。為了表達方便，我們用變數 k 來表示每個二進位輸入的十進位值，並用 $2^k \bmod 35$ 來表示$2^{r_1 r_2 r_3 r_4 r_5 r_6}$ （模 35）。

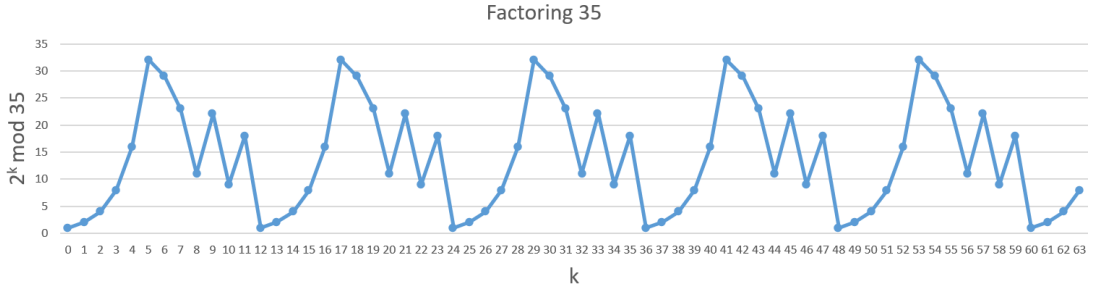


圖 5.5 ：從給定神諭函數 B_f 的 64 個輸出中取樣 64 個點： $\{r_1 r_2$ *右* $_3$ *右* $_4 5$ *號* $r_6 \mid \forall$ *研發* $\in\{0, 1\}$ *為* $1 \le d \le 6\} \rightarrow \{2^{r_1 r_2 r_3 r_4 r_5 r_6}$ (模 35) $\mid \forall$ *研發* $\in\{0, 1\}$ *為* $1 \le d \le 6\}$。

給定神諭函數 B_f 中儲存的隱藏模式和資訊是訊號旋轉回其*第一個*訊號（輸出為 1）*六次*。其訊號循環回到其*第二個*訊號（輸出為 2）*六次*。其訊號循環回到其*第三*訊號（輸出為 4）*六次*，其訊號循環回到其*第四*訊號（輸出為 8）*六次*。其訊號循環回到其*第五*訊號（輸出 16）*五次*，其訊號循環回其*第六*訊號（輸出 32）*五次*。其訊號循環回到其*第七*訊號（以 29 輸出）*五次*，其訊號循環回到其*第八個*訊號（以 23 輸出）*五次*。其訊號循環回到其*第九個*訊號（輸出為 11）*五次*，其訊號循環回到其*第十個*訊號（輸出為 22）*五次*。其訊號循環返回其*第十一個*訊號（輸出 9）*五次*，其訊號循環返回其*第十二*訊號（輸出 18）*五次*。這表示每六十四個時間單位有 $(5\frac{4}{12})$個訊號週期，訊號的頻率 f 等於 $(5\frac{4}{12})$ 。

由於圖 5.5 中訊號的周期 r 是訊號頻率 f 的*倒數*，因此訊號的$\frac{1}{\frac{5\frac{4}{12}}{64}}$周期 r = = $\frac{1}{\frac{\frac{64}{12}}{64}} = \frac{64}{\frac{64}{12}} 12 / 1 = 12$ 。 圖 5.5 滿足 $B_f(r_1{}^0 r_2{}^0 r_3{}^1 r_4{}^1 r_5{}^0 r_6{}^0) = 2^{r_1^0 r_2^0 r_3^1 r_4^1 r_5^0 r_6^0} (\text{mod } 35) = 2^{12} (\text{mod } 35) = 4096 (\text{mod } 35) = 1 (\text{mod } 35)$ ，因此 B_f *中訊號的*周期 $r = 12$ 為相當於 2 模 35 的$2^{r_1 r_2 r_3 r_4 r_5 r_6}$階 $r = 12$ 。（模 35）。因為 $r = 12$ 是偶數且小於 35，所以根據**引理 5-2** ，我們使用歐幾里德演算法來計算 gcd(35, $2^{\frac{12}{2}} + 1$) 和 gcd(35,$2^{\frac{12}{2}} - 1$)。這意味著 $N = 35$ 的兩個重要因子分別為 *5* 和×7 。

因為（ $\frac{1}{\frac{5\frac{4}{12}}{64}} = \frac{1}{\frac{\frac{64}{12}}{64}} == \frac{64}{\frac{64}{12}} 12 / 1)$ 是一個有理數並且是一個整數，我們利用圖 5.2 中的連分數 al 演算法 如果 $c = 12$ 且 $d = 1$ *，則*確定 (c / d) *的連分數表示*以及相應的收斂，以解釋連分數 al 演算法在實際應用中的工作原理。從第一次執行語句 S_0 到語句 S_2 ，得到 $i = 1$ 、 $q[1] = c / d = 12 / 1 = 12$ 和 $r = 12 (\text{mod } 1) = 0$。) 轉換其整數和小數部分且不反轉其小數部分，

$$\frac{12}{1} = 12 + \frac{0}{1} = 12 \text{ 。} \qquad (5.16)$$

r 的值等於 0，因此從第一次執行語句 $S_{3\ 開始}$，它會傳回 *true* 。因此，接下來，從第一次執行語句 $S_{4\ 開始}$，答案是 (12 / 1) 的連分數表示

$$\frac{12}{1} = (q\,[1] = 12) = 12.\ (5.\ 17)$$

接下來，從第一次執行語句 $S_{5\ 開始}$，終止連分數 a 演算法的執行。對於有理數 (12 / 1)，第一個收斂是 $(q\,[1]) = 12 =$ ，$\frac{12}{1}$它最接近($\frac{1}{5\frac{4}{\frac{12}{64}}} = \frac{1}{\frac{64}{\frac{12}{64}}} = \frac{64}{\frac{64}{12}} = \frac{12}{1}$)並且實際上等於$\frac{12}{1}$。這就是說，第一次收斂$(q\,[1]) = 12 = \frac{12}{1}$等於週期 $r = \frac{r}{1}$。因此，我們得到週期 r 等於分子 12 的第一個收斂。分子 $r = 12$ 第一個收斂的值小於 N= 35 ，因此分子 $r = 12$ 等價於 命令 $r = 12$ of 2 mod 35 滿足 $2^{12} = 1 \pmod{35}$。

5.9 求 5 模 33 的階數和 33 的質因數分解

我們**想要**搜尋 $N = 33$ 的*質因數*分解。 = 5 和 $N = 33$ 和為 1（一）。這就是說 **X** = 5 與 $N = 33$ *互質*。 命令 r 為 5 模 33 滿足 r 33. 表示≤$N = 33$ 的位數有 6 位長，我們只需要使用 6 位來編碼 r 的值。

計算 命令 r 對 5 模 33 相當於計算週期 r 給定的神諭函數 C_f: { $r_1 r_2$ *右* $_3$ *右* $_{45}$ *號* $r_6 | \forall$ *研發* $\in \{0, 1\}$ 為 $1 \le d \le 6\} \rightarrow \{ 5^{r_1 r_2 r_3 r_4 r_5 r_6}$（模 33）$| \forall$ *研發* $\in \{0, 1\}$ 為 $1 \le d \le 6\}$。 C_f的周期 r 滿足 $C_f(r_1 r_2 r_3 r_4 r_5 r_6) = C_f(r_1 r_2$ *右* $_3$ *右* $_{45}$ *號* $r_6 + r)$ 到任兩個輸入 ($r_1 r_2$ *右* $_3$ *右* $_{45}$ *號* r_6) 和 ($r_1 r_2$ *右* $_3$ *右* $_{45}$ *號* $r_6 + r$)。 C_f的前 *20 個輸出*，每個輸入來自 $r_1{}^0 r_2{}^0 3_0 4^0 5_0 r_6{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 5_1 r_6{}^1$ 依序為 1, 5, 25, 26, 31, 23, 16, 14, 4, 20, 1, 5, 25, 26, 31, 23, 16, 14, 4 和 *20* 。$_f$分別為 1、5、25、26、31、23、16、14、4、20、1、5、25、26、31、23、16、14、4 和 20 。個數隨後為 1、5、25、26、31、23、16、14、4、20、1、5、25、26、31、23、16、14、4、20、1、5、25、26、31、23、16、14、4、20、1、5、25 26 . C_f的頻率 f 等於每六十四個輸出的週期數。 C_f的周期 r 是 C_f的頻率 f *的倒數*。因此，我們得到 $r = \frac{r}{1} = \frac{1}{\frac{f}{64}} = \frac{64}{f}$和 $r \times f = 64 \times 1 = 64$。

另一方面，我們將 $C_{f\ 的輸入域}$視為時域，輸出視為訊號。計算 命令 r 對 5 模 33 相當於確定了時域（輸入域）訊號的周期 r 和頻率 f 。 $r_1{}^0 r_2{}^0$ 每個輸入的前二

十個輸出 $3_0 4^0 5_0 r_6{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 5_1 r_6{}^1$ 接著是 1、5、25、26、31、23、16、14、4、20、1、5、25、26、31、23、16、14、4 和 20 。*分別為* 1、5、25、26、31、23、16、14、4、20、1、5、25、26、31、23、16、14、4 和 20 。 5、25、 26、 31、 23、 16、 14、 4、 20、 1、 5、 25、 26、 31、 23、 16、 14、 4、 20、 1、 5、 25 和 26 因此，我們將。 64 個輸入值作為對應的 64 個時間單位，將 64 個輸出作為 64 個訊號樣本。每個樣本對 C_f 的輸出進行編碼。輸出*可以取* $1、5、25、26、31、23、16、14、4$ *或* 20 。 $3_0 4^0 5_0 r_6{}^0$ 至 $r_1{}^1 r_2{}^1 3^1 4_1 5_1 r_6{}^1$ 對應於從零到六十三的六十四個時間單位。

我們用圖 5.6 來解釋為什麼要計算 命令 $r5$ 模 33 相當於計算時域(輸入域)訊號的週期 r 和頻率 f 。在圖 5.6 中，橫軸表示時域，即 C_f *的輸入域*。垂直軸表示對 C_f *的六十四個輸出進行編碼的訊號*。為了表達方便，我們用變數 k 來表示每個二進位輸入的十進位值，並用 $5^k \bmod 33$ 來表示$5^{r_1 r_2 r_3 r_4 r_5 r_6}$ （模 33）。

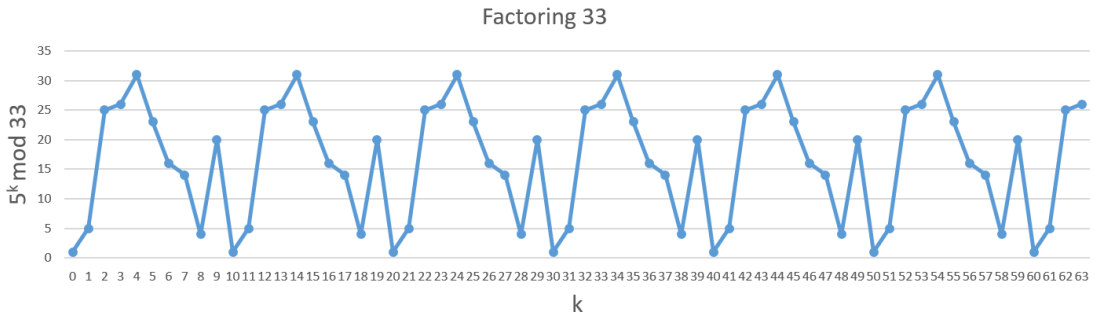


圖 5.6 ：從給定神諭函數 C_f 的 64 個輸出中取樣 64 個點： $\{r_1 r_2$ *右* 3 *右* $4 5$ *號* $r_6 \mid \forall$ *研發* $\in\{0, 1\}$ *為* $1 \le d \le 6\} \rightarrow \{5^{r_1 r_2 r_3 r_4 r_5 r_6}$ (模 33) $\mid \forall$ *研發* $\in\{0, 1\}$ *為* $1 \le d \le 6\}$。

給定神諭函數 C_f 中儲存的隱藏模式和訊息是訊號旋轉回其*第一個*訊號(輸出為 1）*七次*。其訊號循環回到其*第二個*訊號（輸出為 5）*七次*。其訊號循環回到其*第三*訊號(輸出為 25）*七次*，其訊號循環回到其*第四*訊號(輸出為 26）*七次*。其訊號循環回到其*第五*訊號(以 31 輸出）*六次*，其訊號循環回到其*第六*訊號(以 23 輸出）*六次*。其訊號循環回到其*第七*訊號（輸出 16）*六次*，其訊號循環回其*第八*訊號（輸出 14）*六次*。其訊號循環回到其*第九個*訊號（輸出為 4）*六次*，其訊號循環回到其*第十個*訊號（輸出為 20）*六次*。這表示每六十四個時間單位有 $(6\frac{4}{10})$個訊號週期，且訊號的頻率 f 等於 $(6\frac{4}{10})$ 。

訊號的周期 r *是*訊號頻率 f 的*倒數*，所以訊號的$\frac{1}{6\frac{4}{10}}$周期 $r = = \frac{1}{\frac{64}{10}} = \frac{64}{\frac{64}{10}} 10 / 1 =$

10 。 圖 5.6 滿足 $C_f(r_1{}^0 r_2{}^0 r_3{}^1 r_4{}^0 r_5{}^1 r_6{}^0) = 5^{r_1^0 r_2^0 r_3^1 r_4^0 r_5^1 r_6^0} (\text{mod } 33) = 5^{10}$ (mod 33) = 9765625 (mod 33) = 1 (mod 33) ，因此 $C_{f\ 中訊號的}$週期 $r = 10$ 為相當於 5 模 33 的階$5^{r_1 r_2 r_3 r_4 r_5 r_6} r = 10$ 。 （模 33）。由於 $r = 10$ 是偶數且小於 33，根據 **引理 5-2** ，我們使用歐幾里德演算法來計算 gcd(33, $5^{\frac{10}{2}} + 1$) 和 gcd(33,$5^{\frac{10}{2}} - 1$)。這顯示 $N = 33$ 的兩個重要因子分別為 *3* 和×11 。

由於 ($\frac{1}{6\frac{4}{\frac{10}{64}}} = \frac{1}{\frac{64}{\frac{10}{64}}} == \frac{64}{\frac{64}{10}} 10 / 1$) 是一個有理數且是整數，我們使用圖 5.2 中的連分數 al 演算法 確定 (c / d) 如果 $c = 10$ 且 $d = 1$ 時的連分式表示以及相應的收斂性，以解釋連分式 al 演算法在實際應用中的工作原理。從第一次執行語句 S_0 到語句 S_2 ，得到 $i = 1$， $q[1] = c / d = 10 / 1 = 10$ 和 $r = 10 \text{ (mod } 1) = 0$。) 轉換其整數和小數部分且不反轉其小數部分，

$$\frac{10}{1} = 10 + \frac{0}{1} = 10 \text{ 。} \qquad (5.18)$$

r 的值等於 0，因此從第一次執行語句 $S_{3\ 開始}$，它會傳回 *true* 。因此，接下來，從第一次執行語句 $S_{4\ 開始}$，答案就是 (10 / 1) 的連分數表示

$$\frac{10}{1} = (q[1] = 10) = 10. \quad (5.19)$$

接下來，從第一次執行語句 $S_{5\ 開始}$，終止連分數 a 演算法的執行。對於有理數 (10 / 1)，第一個收斂是 ($q[1]$) = 10 = ，$\frac{10}{1}$它最接近($\frac{1}{6\frac{4}{\frac{10}{64}}} = \frac{1}{\frac{64}{\frac{10}{64}}} = \frac{64}{\frac{64}{10}} = \frac{10}{1}$)並且實際上等於$\frac{10}{1}$。這意味著第一個收斂($q[1]$) = 10 =$\frac{10}{1}$等於週期 $r = \frac{r}{1}$。因此，我們得到週期 r 等於分子 10 的第一個收斂。因為分子 $r = 10$ 第一個收斂的值小於 $N = 33$ ，分子 $r = 10$ 相當於 命令 $r = 10$ of 5 mod 33 滿足 $5^{10} = 1$ (mod 33)。

X 模 *N* 的偶數階的可能性

我們假設整數集合 $\boldsymbol{Z}$ = { …, –3, –2, –1, 0, 1, 2, 3, } 和…自然數集合 $\boldsymbol{Y}$ = {0, 1,

2, 3, }。…概念 $d \mid a$ （唸作「 d ***劃分*** a ”) 表示 $a = q \times d$ 對某一整數 q ， a 是 d 的***倍數***， d *是* a 的除數。例如***， 15 是 1、3、5 和 15 的倍數，1、3、5*** *和 15 是 15* 的約數。整數 a 的非平凡除數也稱為 a 的***因數***。例如，15 的因數（非平凡因數）是 3 和 5。

一個整數 $a > 1$，其唯一約數是平凡約數 a 和 1，稱為***質***數（或更簡單地說，***質數***）。***前六個***質數依序為 2、3、5、7、11 *和* 13 。例如，15 是合數，因為它具有因數 3 和 5 。同樣，整數 0 和所有負整數既不是質數也不是合數。根據 (5.1)，對於給定任何正整數 w 和 N ，存在唯一整數 q 和 r 使得 $0 \le r < N$ 且 $w = q \times N + r$ 。整數 q 是 w 除以 N *的****商數***（結果） 。整數 r 是 w 除以 N 的***餘數***，我們寫成 $r = w \pmod N$。給定任何整數 N *，我們可以將整數分成* N 的倍數和不是 N *的倍數的整數*。透過根據除以 N *時的餘數對* N 的倍數和非倍數進行分類，我們可以得到該劃分的細化。

根據它們對 N *的餘數*，該整數可以分為 N *個*等價類。等效***類模*** N 包括整數 w 是

$$[w]_N = \{ w + q \times \text{氮} : \ q \in \boldsymbol{Z} \}。 \quad (5.20)$$

例如，$[2]_5 = \{ \ldots, 2, 7, 12, 17, \ldots \}$。如果 w 模 N 的餘數與 a 模 N 的餘數相同，那麼我們可以說寫*一個* $\in [w]_N$ *與* $a = w \pmod N$ *)*相同。所有此類等價類的集合是

$$\boldsymbol{Z}_N = \{ [w]_N : 0 \le w \le \text{氮} - 1 \} = \{0, 1, 2, 3, \ldots N-1\}。 \quad (5.21)$$

在（5.21）中，我們用 0 來表示 $[0]_N$ ，我們用 1 來表示 $[1]_N$ ，我們用 2 來表示 $[2]_N$ 等等，我們用 apply $N-1$ 代表 $[N-1]_N$.我們用它的最小非負元素來表示每個類別。

因為兩個整數的等價類唯一地確定了它們的和、積或差的等價類，所以我們可以輕鬆地定義 $\boldsymbol{Z}_N$ *的加法、乘法和減法運算*。這就是說，如果 $c = c^1 \pmod N$ 且 $d = d^1 \pmod N$，則

$$c + d = c^1 + d^1 \pmod N, \ c \times d = c^1 \times d^1 \pmod N \text{ 和 } c - d = c^1 - d^1 \text{ (模 } N) \quad (5.22)$$

因此，我們表示以 N *為模的加法、乘法和減法*，定義為 $+_N$ 、 $\times_N$ 和 $-_N$ ，如下所示：

$$[c]_N +_N [d]_N = [c + d]_N, \ [c]_N \times_{\text{氮}} [d]_N = [c \times d]_N \text{和} [c]_N -_{\text{氮}} [d]_N = [c - d]$$

$_N$ 。 (5.23)

（5.23）中的加法模 N 的定義，我們定義***加法群模*** N 為 $(z_N, +_N)$。我們使用**引理 5-7** 來證明系統 $(\boldsymbol{Z}_N, +_N)$ 是有限阿貝爾群。

引理 5-7 ：系統 $(\boldsymbol{Z}_N, +_N)$ 是有限阿貝爾群。

證明：

$\boldsymbol{Z}_N$ 中的任兩個元素 $[c]_N$ 和 $[d]_N$ ，從(5.20) 到 (5.23)，我們得到 $0 \le c \le$ 氮 $-1, 0 \le d \le$ 氮 -1 和 $[c]_N +_N [d]_N = [c+d]_N$ 。如果 $0 \le c+d \le$ 氮 -1，則 $[c]_N +_N [d]_N = [c+d]_N$ 是 $\boldsymbol{ZN}$ 中的元素。如果 $N \le c+d \le 2\times$ 氮 -2、則 $[c]_N +_N [d]_N = [c+d]_N = [c+d-N]_N$ 。因為 $0 \le c+d-$ 氮 $\le$ 氮 2、它是 $-\boldsymbol{Z}_N$ 中的一個元素。這意味著系統 $(\boldsymbol{Z}_N, +_N)$ 是封閉的。

$\boldsymbol{Z}_N$ 中的任三個元素 $[c]_N$ 、 $[d]_N$ 和 $[e]_N$ ，從 (5.23) 可以得到 $([c]_N +_N [d]_N) +_N [e]_N = ([c+d]_N) +_N [e]_N = [(c+d)+e]_N = [c+(d+e)]_N = [c]_N +_N ([d+e]_N) = [c]_N +_N ([d]_N) +_N [e]_N$ 。這顯示系統 $(z_N, +_N)$ 滿足 $+_N$ 的結合律。

$\boldsymbol{Z}_N$ 中的任兩個元素 $[c]_N$ 和 $[d]_N$ ，從 (5.23) 可以得到 $[c]_N +_N [d]_N = [c+d]_N = [d+c]_N = [d]_N +_N [c]_N$.這意味著系統 $(\boldsymbol{Z}_N, +_N)$ 滿足 $+_N$ 的交換律。

系統的單位元 $(\boldsymbol{Z}_N, +_N)$ 是 $[0]_N$ ，因為對於 $\boldsymbol{Z}_N$ 中的任何元素 $[c]_N$ ，從 (5.23) 可以得到 $[c]_N +_N [0]_N = [c+0]_N = [c]_N = [0+c]_N = [0]_N +_N [c]_N$. $\boldsymbol{Z}_N$ 中任意元素 $[c]_N$ 的加法逆元為 $[N-c]_N$ 因為 $[c]_N +_N [$氮 $-c]_N = [c+N-c]_N = [N]_N = [0]_N$ 。系統中的元素數量 $(\boldsymbol{Z}_N, +_N)$ 為 N ，因此它是有限的。因此，從上面的陳述，我們立即推斷系統（ $\boldsymbol{ZN}$ 、 $+_N$ ）是一個有限交換群。■

集合 $\boldsymbol{Z}_N^*$ 是 $\boldsymbol{Z}_N$ 中與 N 互質的元素的集合，且是

$$\boldsymbol{Z}_N^* = \{ [w]_N \in \boldsymbol{ZN} : \gcd(w, N) = 1 \} 。 \quad (5.24)$$

因為 $[w]_N = \{ w + q \times$ 氮： $q \in \boldsymbol{Z} \}$ 和 $\gcd(w, N) = 1$，我們有 $\gcd(w + q \times N, N) = \boldsymbol{1}$ 、 $14]_{15} \}$。使用（5.23）中模 N 乘法的定義，我們將模 N 乘法群表示為 $(\boldsymbol{Z}_N^*, \times_N)$。我們利用**引理 5-8** 證明系統 $(\boldsymbol{Z}_N^*, \times_N)$ 是有限阿貝爾群。

引理 5-8 ：系統 $(\boldsymbol{Z}_N^*, \times_N)$ 是有限阿貝爾群。

證明：

$\boldsymbol{Z}_N^*$中的任兩個元素$[c]_N$和$[d]_N$，從(5.20) 到 (5.23)，我們得到 $0 \le c \le$ 氮 $-1, 0 \le d \le$ 氮 -1、$\gcd(c, N) = 1, \gcd(d, N) = 1$ 且$[c]_N \times_{氮} [d]_N = [c \times d]_N$。因為 $\gcd(c, N) = 1$ 且 $\gcd(d, N) = 1$，所以我們有 $\gcd(c \times d, N) = 1$. 這表示$[c]_N \times_{氮} [d]_N = [c \times d]_N$是$\boldsymbol{Z}_N^*$中的元素。因此，系統$(Z_N^*, \times_N)$是封閉的。

$\boldsymbol{Z}_N^*$中的任三個元素$[c]_N$、$[d]_N$和$[e]_N$，從 (5.23) 可以得到 $([c]_N \times_{氮} [d]_N) \times_N [e]_N = ([c \times d]_N) \times_N [e]_N = [(c \times d) \times e]_N = [c \times (d \times e)]_N = [c]_N \times_N ([d + e]_N) = [c]_N \times_N ([d]_N) \times_N [e]_N$）。這就是說，系統$(Z_N^*, \times_N)$滿足的結合律×。

$\boldsymbol{Z}_N^*$中的任兩個元素$[c]_N$和$[d]_N$，由 (5.23) 可以得到$[c]_N \times_{氮} [d]_N = [c \times]_N = [\mathrm{d} \times c]_N = [d]_N \times_{氮} [c]_N$.這顯示系統$(Z_N^*, \times_N)$滿足的交換律×。

$\boldsymbol{Z}_N^*$中的任何元素$[c]_N$，根據 (5.23)，我們有$[c]_N \times_{氮} [1]_N = [c \times 1]_N = [c]_N = [1 \times c]_N = [1]_N \times_{氮} [c]_N$.這表示系統的單位元（ $\boldsymbol{Z}_N^*, \times_N$ ）是$[1]_N$。

$\boldsymbol{Z}_N^*$中的*任何*元素$[c]_N$滿足 $\gcd(c, N) = \mathbf{1}$。使得$[c]_N \times_{氮} [c^{-1}]_N = [c \times c^{-1}]_N = [1]_N = [c^{-1} \times c]_N = [c^{-1}]_N \times_{氮} [c]_N$.系統中的元素數量 $(\boldsymbol{Z}_N^*, \times_N)$ 小於 N，因此它是有限的。因此，根據上面的陳述，我們立即推導出系統 $(\boldsymbol{Z}_N^*, \times_N)$ 是一個有限交換群。■

$\boldsymbol{Z}_N^*$的大小稱為***歐拉 phi 函數*** $\phi(N)$滿足下式

$$\phi(N) = N \times \left(\prod_{p|N}\left(1 - \frac{1}{p}\right)\right), \quad (5.25)$$

在哪裡 p 遍歷除 n *的所有質數*（包括 N 本身，如果 N 是質數）。 例如，因為 15 的質因數是 3 和 5，所以我們得到$\phi(15) = 15 \times (1 - (1/3)) \times (1 - (1/5)) = 15 \times (2/3) \times (4/5) = 8$。 $[11]_{15}$、$[13]_{15}$、$[14]_{15}\}$。若 p 是質數，則 p 本身是唯一的質因數。因此，根據 (5.25)，我們得到$\phi(p) = p \times (1 - (1/p)) = p - 1$. 唯一小於 p^a 且不與 p^{a} *互質的整數是* p 的倍數： $p, 2 \times p, \ldots, (pa^{-1} - 1) \times p$，從中我們推論

$$\phi(p^a) = (p^a - {}^{1}) - (帕^{-1} - 1) = pa - 帕^{-1} = p^{a-1} \times (p - 1)。 \quad (5.26)$$

此外，如果 c 和 d 互質，則$\phi(c \times d)$ 滿足下列方程

$$\phi(c \times d) = \phi(c) \times \phi（四）。 \quad (5.27)$$

另一方面，當 N *是奇質數* p 的冪時， $N = p^a$ 。事實證明， $\mathbf{Z}_N^* = \mathbf{Z}_{p^a}^*$ 是一個*循環*群，也就是說， $\mathbf{Z}_{p^a}^*$ 中有一個元素 h 生成 $\mathbf{Z}_{p^a}^*$ ，任何其他元素 y 都可以寫成 $y = h^m \pmod{N} = h^m \pmod{p^a}$ 對於某個非負整數 m 。我們用**引理 5-9** 和**引理 5-10** 來解釋找出不等於 N *的偶數階的可能性* $-X$ 模 N 的 1 。

引理 5-9 ：我們假設 p 是奇質數，2^b *是* 2 *除法* (p^a)的最大冪。ϕ然後，以恰好二分之一的機率 2^b *除以* $\mathbf{Z}_{p^a}^*$的隨機選取元素的階模 p^a 。

證明：

因為 p 是奇質數，所以由 (5.26) 我們得到$\phi(p^a) = p^{a-1} \times (p-1)$ 是偶數。因為$\phi(p^a)$ 是偶數且 2^b 除$\phi(p^a)$，我們得到 $b \geq 1$. 由於 $\mathbf{Z}_{p^a}^*$是一個*循環*群，因此 $\mathbf{Z}_{p^a}^*$中存在一個元素 h ，它產生 $\mathbf{Z}_{p^a}^*$ ，任何其他元素 X 都可以寫成 $X = h^m \pmod{p^a}$ 對於某些 m *在* 1 *到*(p^a)範圍內，即$\phi\mathbf{Z}_{p^a}^*$的大小。設 r *為* h^m 模 p^a 的階並考慮兩種情況。第一種情況是 m 為奇數時。由於 h *與* $(p^a)\phi$*互質*，且(p^a) *是* $\mathbf{Z}_{p^a}^*$的大小，因此$\phi(p^a)$ *是滿足* $= 1 \pmod{p^a}$的最小值$h^{\phi(p^a)}$。因為 $(h^m)^r = h^{m \times r} = 1 \pmod{p^a}$，我們推論$\phi(p^a)$ 除 $(m \times)$ 。由於 m 是奇數， $\phi(p^a)$ 是偶數，因此 2^b 整除$\phi(p^a)$ 且$\phi(p^a)$ 除 $(m \times r)$ 和 2^b 除 $(m \times r)$，我們推論 2^b 整除 r 。第二種情況是 m 為偶數時。因為 h *與* p^a *互質*，且 m 是偶數，所以我們推論 $h^{m/2}$ 模 p^a *與* p^a *互質*。因此，我們有 $(h^{m \times \phi(p^a)/2}) = (h^{\phi(p^a)})^{m/2} = (1)^{m/2} = 1 \pmod{p^a}$。因為 r *是* h^m 模 p^a 的階數，它是滿足 $(h^m)^r = h^m$ 的最小值 $\times r = 1 \pmod{p^a}$，我們推論 r 除以 $(\phi(p^a)/2)$ 且 r 小於 2^b ，即 2 除$\phi(p^a)$ 的最大冪次方。因此，我們推論 2^b 不能整除 r 。

m 的值在 1 到(p^a)範圍內，是偶數，且是$\phi\mathbf{Z}_{p^a}^*$的大小，所以我們可以將 $\mathbf{Z}_{p^a}^*$分成兩個大小相等的集合。第一個相同大小的集合可以寫成 $h^m \pmod{p^a}$，其中 m 是奇數，其中 2^b 除以 h^m 模 p^a 的階數 r 。第二組大小相等，可以寫成 $h^m \pmod{p^a}$，其中 m 是偶數，其中 2^b 不能除以 h^m 模 p^a 的階數 r 。因此，整數 2^b *以機率* $(1/2)$ *整除*隨機選取的元素 $\mathbf{Z}_{p^a}^*$*的*階數 r ，而以機率(1 / 2)則不能整除。■

引理 5-10 ：我們假設 $N = p_1^{a_1} \times \ldots \times p_m^{a_m}$是奇複合正整數的質因數分解。令 X *從* $\mathbf{Z}_N^*$中均勻隨機選擇，並令 r *為* X 模 N 的階數。則 P (r 是偶數且 $X^{r/2} \neq -1 \pmod{N}) \geq 1 - (1/2^m)$。

證明：

我們證明 P(r 是奇數或 $X^{r/2} = -1 \pmod N$) ≤1 / 2 ^{m} 。根據中國剩餘定理，從 $\boldsymbol{Z}_N{}^*$中均勻隨機選擇 X 相當於從 獨立且均勻隨機選擇$\boldsymbol{Z}^*_{p_k^{a_k}} X_k$ ， 並滿足 $X = X_k \pmod{p_k^{a_k}}$ 對於 1≤ k≤ *米*。設 r_k *為* X_k 模 ( )$p_k^{a_k}$的階數。設為 2 除以$2^{b_k} r_k$的最大冪，2 $^{b\ 為\ 2\ 除以}$ r 的最大冪。因為 X *與* N 互質（ $p_1^{a_1} \times \ldots \times p_m^{a_m}$) , ϕ(N) = ϕ($p_1^{a_1} \times \ldots \times$) 是$p_m^{a_m}\boldsymbol{Z}_N{}^*$的大小，它是== $X^{\phi(p_1^{a_1} \times \ldots \times p_m^{a_m})} 1 \pmod N$)的最小值$X^{\phi(N)}$。由於 r *是* X 模 N 的階數，即$X^r = 1 \pmod N$) 的最小值，因此我們有 r = $\phi(p_1^{a_1} \times \ldots \times p_m^{a_m}) = \phi(p_1^{a_1}) \times \ldots \times \phi(p_m^{a_m})$。因為對於 1， ≤$X_k$ *與* ( )互質$p_k^{a_k}$ $k \le m$, ϕ( ) 是滿足$p_k^{a_k} = 1 \pmod{(\ )p_k^{a_k}}$的最小值$X_k^{\phi(p_k^{a_k})}$的$\boldsymbol{Z}^*_{p_k^{a_k}}$大小。由於 r_k *是* X_k 模數 ( )$p_k^{a_k}$的階數，因此是滿足 $X_k{}^r$ 的最小值 =1 (mod ($p_k^{a_k}$))，我們有 $_{rk}$ = (ϕ)$p_k^{a_k}$為 1≤ k≤ *米*。因為 $r = \phi(p_1^{a_1}) \times \ldots \times \phi(p_m^{a_m})$ 且 $_{rk}$ = (ϕ)$p_k^{a_k}$對於 1≤ $k \le m$ ，我們≤推斷 rk 除以 $_1$ k≤ *米*。我們將證明，要讓 r *為*奇數或 $X^{r/2} = -1 \pmod N$*)*， b_k 必須採用與 1≤相同的值 k≤ *米*。結果如下，根據**引理 5-9** ，這種情況發生的機率最多為 (1 / 2) ×(1 / 2)× … ×(1 /2) = 1 / 2 ^{m} 。

我們考慮第一種情況是 r 為奇數時。因為 r_k 除以 *1*≤ $k \le m$ ，我們推論 rk *是*奇數。因為$2^{b_k} r_k$除以 1≤ $k \le m$ ，我們得到 $b_k = 0$ 對於 1≤ k≤ *米*。第二種情況是當 r 為偶數且 $X^{r/2} = -1 \pmod N$) 時。這就是說 $X^{r/2} = N - 1 = p_1^{a_1} \times \ldots \times p_m^{a_m} - 1$. 因此，我們有 $X^{r/2} = N - 1 = p_1^{a_1} \times \ldots \times p_m^{a_m} - 1 = -1 \pmod{(\ p_k^{a_k})}$。因此我們得到 r_k 不能整除 (r / 2)。因為 r_k 除以 *1*≤ $k \le m$ ，我們必須有 $b_k = b$ 為 1≤ k≤ *米*。由於 P(r 是偶數且 $X^{r/2} \ne -1 \pmod N$) + P(r 為奇數或 $X^{r/2} = -1 \pmod N$) = 1 且 P(r 為奇數或 $X^{r/2} = -1 \pmod N$) ≤1 / 2 ^{m} ，我們有 P(r 是偶數且 $X^{r/2} \ne -1 \pmod N$) ≥1 –(1 / 2 ^{m})。■

5.1 1 公鑰密碼學與 RSA 密碼系統

消費者想要在網路上購買東西。他希望透過網路傳輸他的信用卡號碼，以便只有提供他所購買產品的公司才能收到該號碼。一個*密碼學* 互聯網上的*協定*或*密碼系統可以實現這種私密通訊*。有效的密碼系統使想要相互通信的兩方變得容易，但使竊聽者很難竊聽對話內容。

一類特別重要的密碼系統是*公鑰密碼系統*。在公鑰密碼系統中，瑪麗想要向她的朋友發送訊息並接收她的朋友發送的訊息。她必須先產生兩個*密鑰*。一個是公鑰 P ，另一個是秘密金鑰 S 。瑪麗產生密鑰後，她宣布或發佈公鑰，以便任何人都可以存取該公鑰。

約翰是瑪莉的好朋友。他想傳一封私人訊息給瑪麗。因此，約翰首先獲得瑪麗的公鑰 P *的副本*。然後，他將要傳送給 Mary 的私人訊息進行加密，並利用 Mary 的公鑰 P 來完成加密。由於公鑰和編碼訊息是竊聽者可用的唯一訊息，因此竊聽者不可能恢復該訊息。然而，瑪麗擁有竊聽者無法獲得的秘密密鑰 S。她使用密鑰 S 解密加密訊息並獲得*原始*訊息。這種稱為解密的轉換與加密相反，允許瑪麗恢復約翰的私人訊息。

使用最廣泛的公鑰密碼系統是 **RSA** 密碼系統，以其創作者 Rivest、Shamir 和 Adleman 的名字縮寫命名為 **RSA** 。 **RSA 密碼系統**的假定安全性是基於在數位計算機上進行因式分解的明顯困難。現在 Mary 希望產生用於 **RSA** 密碼系統的公鑰和私鑰。她使用以下過程來產生它們：

(1) 選擇兩個大質數 p 和 q 。
(2) 計算乘積 $N = p \times q$.
(3) 隨機選取一個小的奇數 e ，它與 $\phi(N) = (p-1) \times (q-1)$。
(4) 計算 d ，*即* e 的乘法逆元，模 $\phi(N)$。
(5) 公鑰是 $P = (e, N)$ 對。
(6) 密鑰是 $S = (d, N)$ 對。

現在 John 使用公鑰 (e, N) 加密訊息 M 傳送給 Mary。我們假設訊息 M 只有 $\log_2 N$ 位，因為可以透過將 M 分成最多 $\log_2 N$ 位的區塊然後單獨加密這些區塊來加密更長的訊息。單一區塊的加密過程是計算：

$$E(M) = Me \pmod{N}。 \quad (5.28)$$

$E(M)$ *是* John 發送給 Mary 的訊息 M 的加密版本。 Mary 能夠使用她的金鑰 $S = (d, N)$ 快速解密該訊息，只需將加密訊息提高 d 次方即可：

$$D(E(M)) = M^{e \times d} (\text{mod } N) = M (\text{mod } N)。 \quad (5.29)$$

RSA 密碼系統如何被破解？答案是，如果我們能夠有效地將一個大合數 N *分解*為兩個大質數 p 和 q *的產生式*。然後，我們可以提取 p 和 q 。這意味著我們可以有效地計算 $\phi(N) = (p-1) \times (q-1)$。接下來，我們可以有效地計算 d ，即 e 的乘法逆元，模 $\phi(N)$。因此，我們可以完全確定密鑰 (d, N) 。因此，如果分解大數很容易，那麼破解 RSA 密碼系統也會很容易。

5.12 實現三個量子位元的受控交換閘

一（ 8 ×8) 矩陣 ***CSWAP*** 及其共軛轉置$\overline{\boldsymbol{CSWAP}}$分別為

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{和} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} 。 \quad (5.30)$$

因為 ***CSWAP*** × $\overline{\boldsymbol{CSWAP}}$ 等於 (8 ×8) 辨識矩陣和 $\overline{\boldsymbol{CSWAP}}$ × ***CSWAP*** 等於 (8 ×8) 辨識矩陣，矩陣 ***CSWAP*** 和矩陣$\overline{\boldsymbol{CSWAP}}$都是酉矩陣（酉算符）。矩陣 ***CSWAP 是***三個量子位元的 ***CSWAP（受控 SWAP ）閘的***矩陣表示。圖 5.7 中的左圖是具有三個量子位元的 ***CSWAP 閘的第一個圖形電路表示。***量子位|圖 5.7 左圖中最下方的 $C_1>$ 為受控位，量子位$|S_1>$ 頂端與量子位元 |圖 5.7 左圖中間的 $S_2>$ 均為目標位。***CSWAP 閘***的功能是如果控制位元 $|C_1>$等於|1>，則交換兩個目標位元|中所包含的資訊。 $S_1>$ 和 $|S_2>$.否則，它不會交換兩個目標位中包含的資訊$|S_1>$ 和 $|S_2>$.圖 5.7 中的中間圖片是具有三個量子位元的 ***CSWAP 閘的第二個圖形電路表示。圖 5.7 右圖是***利用三個 ***CCNOT 實現 CSWAP 閘***的電路圖 蓋茲。圖 5.7 右圖中，如果控制位$|C_1>$等於|1>，則使用三個 ***CNOT*** 閘實現一個***交換***門來交換兩個目標位元所包含的訊息 $S_1>$ 和 $|S_2>$.否則，它不會實作三個 ***CNOT*** | 門來完成一個***交換***門並交換兩個目標位中包含的信息 $S_1>$ 和 $|S_2>$.

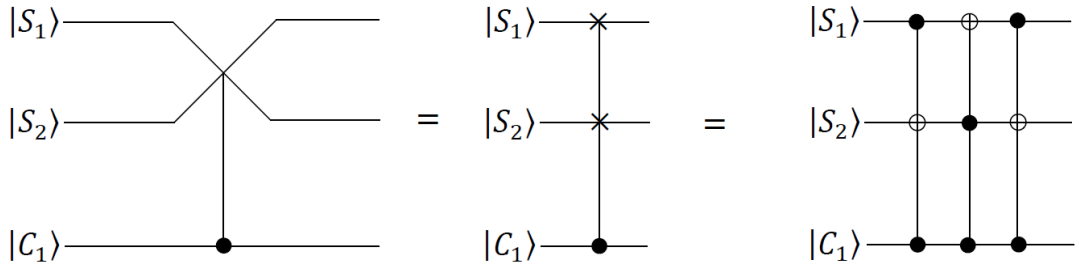


圖 5 .圖 7 ：具有三個量子位元的 **CSWAP 閘**的電路表示。

5.12 .1 實現三個量子位元的受控交換閘的量子程序

在 **IBM Q** Experience 中，它不提供一個用三個量子位元實現 ***CCNOT 閘（Toffoli 閘）的量子指令（操作）。***我們將 **CCNOT** 門分解為*六個* **CNOT** 閘和

1 個量子位元的 *9 個*閘，如圖 5.8 所示。在圖 5.8 中， H 是哈達瑪門， $T=\begin{bmatrix}1 & 0\\ 0 & e^{\sqrt{-1}\times\frac{\pi}{4}}\end{bmatrix}$ 且 $T^{+}=\begin{bmatrix}1 & 0\\ 0 & e^{-1\times\sqrt{-1}\times\frac{\pi}{4}}\end{bmatrix}$。在具有 32 個量子位元的後端*模擬器中*，32 個量子位元之間的 ***CNOT 閘***的連通性沒有限制。

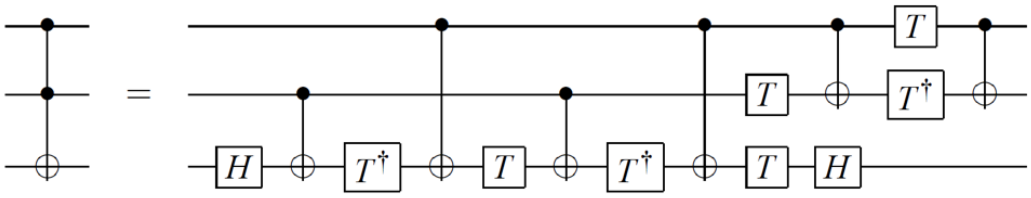


圖 5 . 8：將 **CCNOT** 閘分解為 6 個 **CNOT** 閘和 9 個 1 位閘。

IBM 量子電腦中具有 32 個量子位元的後端*模擬器*中的程式是*第五章*的*第一個範例*，其中我們說明如何編寫量子程式來實現具有 3 個量子位元的 CSWAP 閘。圖 5.9 是清單 5.1 中程式對應的量子電路。為了方便演示，清單 5.1 中的同一行有四個指令。我們使用「指令號」或「行號」來表示清單 5.1 中每個指令的執行順序。

聲明“OPENQASM 2.0；”清單 5.1 第一行中的第一個指令是指出程式是用 Open QASM 2.0 版本寫的。接下來，語句「 include」qelib1.inc」； 」 清單 5.1 第一行中的第二條指令是繼續解析檔案“ q elib1.inc”，就好像該檔案的內容被貼上到 include 語句的位置，其中檔案「 q elib1.inc 」是 **Quantum Experience (QE) 標準標頭**，且路徑是相對於目前工作指定的 目錄。

接下來，語句「 qreg q[3];清單 5.1 第一行中的第三條指令是聲明程式中有 3 個量子位元。在圖 5.9 的左上角，三個量子位元依序為 q[0]、q[1] 和 q[2]。每個量子位元的初始值設定為|0>。我們使用三個量子位元 q [0]、q[1] 和 q[2] 來隨後對第一個目標位元 | 進行編碼。 $S_1>$，第二目標位$|S_2>$ 和控制位元$|C_1>$.

```
1.OPENQASM 2.0； 2. 包含“qelib1.inc”； 3. qreg q[3]； 4. creg c[3];
5.xq[0]; 6.xq[2];
// 用兩個實作圖 5.7 右圖中的第一個 CCNOT 閘
// 控制位元 q[2] 和 q[0] 以及目標位元 q[1]。

7. 障礙 q[0], q[1], q[2]; 8. 總部[1]； 9. cx q[0],q[1]; 10. tdg q[1]；
11.cxq[2],q[1]; 12. tq[1]; 13.cxq[0],q[1]; 14. tdg q[1]；
```

```
15.cxq[2],q[1]; 16. tq[0]; 17. tq[1]; 18.  總部[1]；
19. CX q[2],q[0]; 20 tdg q[0]; 21 tq[2]; 22.cxq[2],q[0];

// 用兩個實作圖 5.7 右圖中的第二個 CCNOT 閘
// 控制位元 q[2] 和 q[1] 以及目標位元 q[0]。

23. 障礙 q[0]，q[1]，q[2]； 24 總部[0]； 25.cxq[1],q[0]; 26. tdg q[0]；
27. CX q[2],q[0]; 28. tq[0]; 29. CX q[1],q[0]; 30. tdg q[0]；
31.cxq[2],q[0]; 32. tq[1]; 33. tq[0]; 34.  總部[0]；
35.cxq[2],q[1]; 36. tdg q[1]； 37. tq[2]； 38.cxq[2],q[1];

// 用兩個實作圖 5.7 右圖中的第三個 CCNOT 閘
// 控制位元 q[2] 和 q[0] 以及目標位元 q[1]。

39. 障礙 q[0]，q[1]，q[2]； 40. 總部[1]； 41.cxq[0],q[1]; 42. tdg q[1]；
43. CX q[2],q[1]; 44. tq[1]； 45.cxq[0],q[1]; 46. tdg q[1]；
47. CX q[2],q[1]; 48. tq[0]; 49. tq[1]; 50.  總部[1]；
51.cxq[2],q[0]; 52. tdg q[0]； 53. tq[2]; 54. CX q[2],q[0];

55. 測量 q[0] -> c[0]；
56. 測量 q[1] -> c[1]；
57. 測量 q[2] -> c[2]；
```

三個量子位元的 ***CSWAP 閘***的程式。

為了方便我們解釋，q[k] ^{0} 代表 $0\le k\le 2$ 是表示 q[k]的值 0，q[k] ^{1} 表示 $0\le k\le 2$ 表示 q[k]的值 1。同樣，為了方便說明，實作 ***CSWAP*** 閘的初始狀態向量如下：

$$|\Phi_0\rangle = |q[2]^0\rangle |q[1]^0\rangle |q[0]^0\rangle = |0\rangle |0\rangle |0\rangle = |000\rangle。$$

然後，語句“ creg c[3];”清單 5.1 第一行的第四條指令是聲明程式中有 3 個經典位元。在圖 5.9 的左下角，三個經典位分別是 c[0]、c[1]和 c[2]。每個經典位的初始值設定為 0。

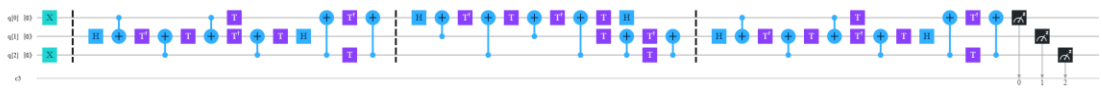


圖 5.9：清單 5.1 程式對應的量子電路。

接下來，這兩個語句 “ xq[0];”和“xq[2]；”清單 5.1 第二行中的第 *5 行*到第 *6* 行在圖 5.9 的量子電路的*第一個*時隙中對量子位元 q[0] 和量子位元 q[2]實施兩個***非閘***。他們實際上都完成了$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \times \begin{pmatrix}1\\ 0\end{pmatrix} = \begin{pmatrix}0\\ 1\end{pmatrix}$= |1>。這表示 q[0]從一種狀態|0>到另一種狀態|1>的轉換以及 q[2]從一種狀態|0>到另一種狀態|1>的轉換已經完成。由於在圖 5.9 中量子電路的*第一個*時隙中沒有量子閘作用於量子位元 q[1]，因此其狀態|0>沒有改變。因此，我們有以下新的狀態向量

$$|\Phi_1\rangle = |q[2]^1\rangle |q[1]^0\rangle |q[0]^1\rangle \quad 。$$

在狀態向量$|\Phi_1\rangle = |q[2]^1\rangle |q[1]^0\rangle |q[0]^1\rangle$，量子位元$|q[2]^1\rangle$為受控位元|的輸入狀態|1> C_1>在圖 5.7 的 ***CSWAP 閘中***。量子位元$|q[1]^0\rangle$是目標位元|的輸入狀態|0> S_2>在圖 5.7 的 ***CSWAP 閘中***。量子位元$|q[0]^1\rangle$是目標位元|的輸入狀態|1>。 S_1>在圖 5.7 的 ***CSWAP 閘中***。接下來，語句“barrier q[0], q[1], q[2];”清單 5.1 第六行的*第七*行實作了一條屏障指令，以防止優化在圖 5.9 中的量子電路的*第二個*時隙中跨其原始碼行重新排序閘。接下來，從清單 5.1 中的指令 *8* 到指令 *22* ，這 15 個語句是“ hq[1];” “cx q[0]，q[1]；”，“tdg q[1]；”，“cx q[2]，q[1]；”，“tq[1]；”，“cx q[0 ”],q[1];”, “tdg q[1];”, “cx q[2],q[1];”, “tq[0];”, “tq[1];”, “ hq [1];”, “cx q[2],q[0];”, “tdg q[0];”, “tq[2];”和“cx q[2]，q[0]；”。他們採用狀態向量$|\Phi_1\rangle = |q[2]^1\rangle |q[1]^0\rangle |q[0]^1\rangle$ 作為其輸入，並使用兩個受控位 q[2] 和 q[0] 以及一個目標位實現第一個 ***CCNOT*** 閘 q[1]從圖 5.9 的*第三個*時隙到*第十五*時隙。由於兩個受控位元 q[2]和 q[0]均為狀態|1>，因此目標位元 q[1]的狀態|0>轉換為狀態|1>。因此，我們有以下新的狀態向量

$$|\Phi_{15}\rangle = |q[2]^1\rangle |q[1]^1\rangle |q[0]^1\rangle \quad 。$$

接下來，語句“barrier q[0], q[1], q[2];”清單 5.1 中的第 23 號指令實作了一條屏障指令，以防止優化在圖 5.9 中的量子電路的*第 16 個*時隙中跨其原始碼行重新排序閘。接下來，從清單 5.1 的指令號 *24* 到指令號 *38* ，*這 15 個語句是*「 hq[0];」、「cx q[1],q[0];」、「tdg q[0] ; ”, “cx q[2],q[0];”, “tq[0];”, “cx q[1],q[0];”, “tdg q[0];”, “cx q [2],q[0];”, “tq[1];”, “tq[0];”, “hq[0];”, “cx q[2],q[1];”, “ tdg q[1];”，“tq[2];”和“cx q[2]，q[1]；”。他們採用狀態向量$|\Phi_{15}\rangle = |q[2]^1\rangle |q[1]^1\rangle |q[0]^1\rangle$ 作為其輸入，並使用兩個控制位 q[2] 和 q[1] 以及一個目標位元來實現第二個 ***CCNOT*** 閘 q[0]從圖 5.9 的*第 17 個*時隙到*第 28 個*時隙。由於兩個受控位元 q[2]和 q[1]都是狀態|1>，因此目標位元 q[0]的狀態|1>轉換為狀態|0>。由此，我們得到如下新的狀態向量

$$|\Phi_{28}> = |q[2]^{1}>|q[1]^{1}>|q[0]^{0}> \quad 。$$

接下來，語句“barrier q[0], q[1], q[2];”清單 5.1 中的*第 39 號*指令實作了一條屏障指令，以防止優化在圖 5.9 中的量子電路的*第 39 個*時隙中跨源線重新排序閘。接下來，從清單 5.1 的指令號 *40* 到指令號 *54* ，這 15 個語句是「 hq[1];」、「cx q[0],q[1];」、「tdg q[1]; ” , “cx q[2],q[1];”, “tq[1];”, “cx q[0],q[1];”, “tdg q[1];”, “cx q [2],q[1];”, “tq[0];”, “tq[1];”, “hq[1];”, “cx q[2],q[0];”, “ tdg q[0];”, “tq[2];”和“cx q[2]，q[0]；”。他們採用狀態向量$|\Phi_{28}> = |q[2]^{1}>|q[1]^{1}>|q[0]^{0}>$ 作為其輸入，並使用兩個控制位 q[2] 和 q[0] 以及一個目標位實現第三個 ***CCNOT*** 閘 q[1]從圖 5.9 的*第 30 個*時隙到*第 40 秒*時隙。由於第一個受控位元 q[2]為狀態|1>且第二受控位元 q[0]為狀態|0>，因此目標位元 q[1]的狀態|1>不變。因此，我們得到以下新的狀態向量

$$|\Phi_{42}> = |q[2]^{1}>|q[1]^{1}>|q[0]^{0}> \quad 。$$

接下來，三個語句“measure q[0] -> c[0];”、“measure q[1] -> c[1];”和“測量 q[2] -> c[2]；”從指令編號 55 到清單 5.1 中的指令編號是測量第一量子位元 q[0]、第二量子位元 q[1]和第三量子位元 q[2]。他們透過覆蓋第一個經典位 c[0]、第二個經典位 c[1] 和第三個經典位 c[2] 來記錄測量結果。在 **IBM 量子電腦**的 32 個量子位元的後端*模擬器中*，我們使用「run」指令來執行清單 5.1 中的程式。測量結果如圖 5.10 所示。從圖 5.10 中，我們得到答案 110 (c[2] = 1 = q[2] = |1>, c[1] = 1 = q[1] = |1> 和 c[0] = 0 = q [0] = |0>)，機率為 100%。因為目標位的輸入狀態$|S_1>$是狀態|1>，目標位元|的輸入狀態$S_2>$是狀態|0>，受控位的輸入狀態$|C_1>$是圖 5.7 中***CSWAP 閘***中的|1> ，兩個目標位元| 所包含的資訊。 $S_1>$ 和 $|S_2>$ 被交換。因此，我們得到了目標位的最終狀態 $|S_1>$ 由 $|q[0]^{0}>$ 編碼為狀態 |0> 以及目標位元 | 的最終狀態由$|q[1]^{1}>$編碼的$S_2>$是狀態|1>，機率為 100%。

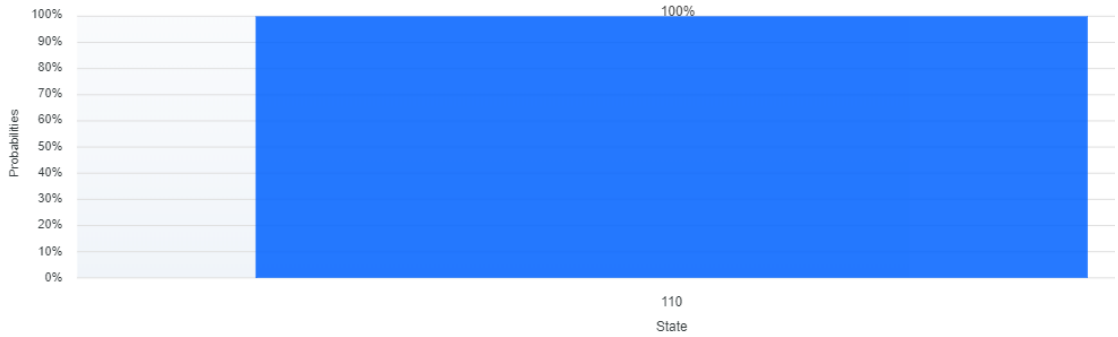


圖 5.10：對清單 5.1 中的程序進行測量後，我們以 100% 的機率得到答案 110。

5.13 Shor 尋序演算法

對於正整數 X 和 N ，*其中 X 的值小於 N 的值且它們的最大公因數為 1*， X 模 N 的階數（週期）為最小正整數 r ，使得 $X^r = 1$（模 N ）。尋序問題是計算給定的 X 和 N *的階數*。在數位計算機上，*沒有已知的演算法*透過使用所需的 O(L) 位元中的資源多項式來解決指定 N 即 L 的位數大於或等於 log 2 (N)的問題來指定問題。在本節中，我們將解釋為何 Shor 的尋序演算法為何是一種高效的尋序量子演算法。

Shor 尋序演算法的量子電路如圖 5.11 所示。 n *個*量子位元$\bigotimes_{k=1}^{n} |p_k^0>$的第一個量子暫存器是 ( )，每個量子位元的初始狀態是 |0> 狀態。量子位$|p_1{}^0>$ 是最高有效位

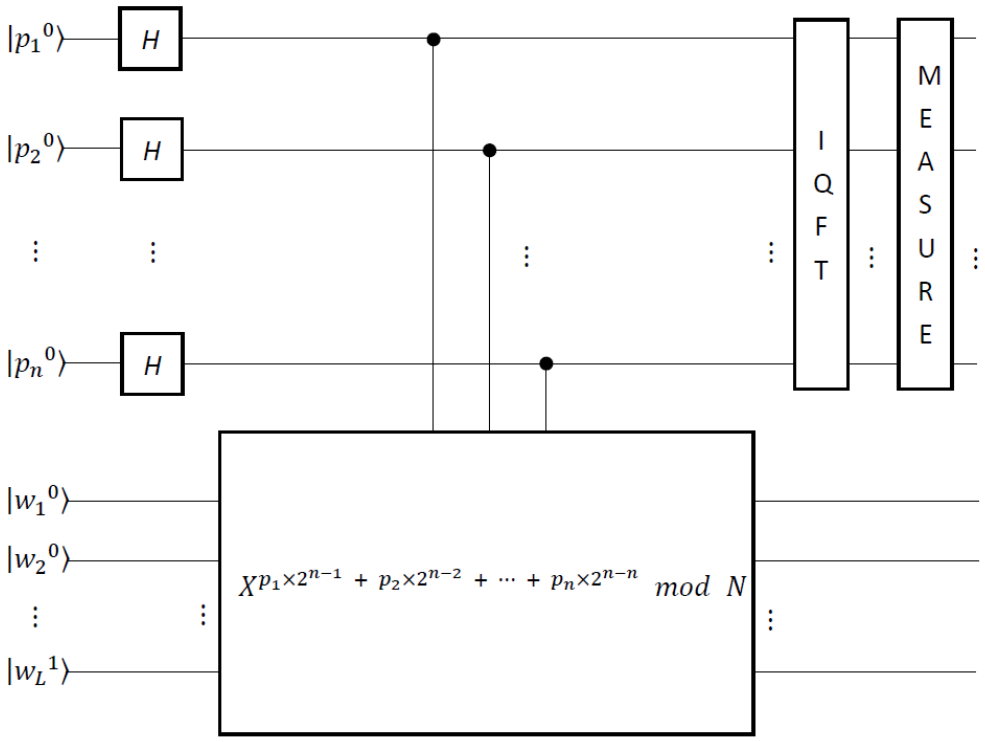


圖 5.11：實現 Shor 尋序演算法的量子電路。

和量子位元$|p_n{}^0>$ 是最低有效位元。由於 X 模 N 的階數小於或等於 N ，因此 n 大於或等於 $\log_2(N)$。其十進制值等於 $p_1 \times 2n^{-1} + p_2 \times 2n^{-2} + p_3 \times 2n^{-3} + \ldots + p_n \times 2n^{-}$ 名詞 L 個量子位元$\bigotimes_{y=1}^{L-1} |w_y^0>$的第二個量子暫存器是 (( ) ⊗($|w_L{}^1>$))。前面每個量子位元的初始狀態（ L –1) 量子位元是|0>狀態。*最低有效*量子位元 ($|w_L{}^1$)>的初始狀態為 |1> 態。量子位$|w_1{}^0>$ 是最高有效位元和量子位元$|w_L{}^1>$ 是

最低有效位元。其十進制值等於 $w_1 \times 2^{公升-1} + w_2 \times 2^{公升-2} + w_3 \times 2^{L-3} + \ldots + p_L \times 2^{公升-L}$。由圖 5.11 可知，初始狀態向量為

$$|\varphi_0\rangle = (\bigotimes_{k=1}^{n} |p_k^0\rangle) \otimes ((\bigotimes_{y=1}^{L-1} |w_y^0\rangle) \otimes (|w_L{}^1\rangle))。 \quad (5.31)$$

由圖 5.11 可知，初始狀態向量|(5.31) 中的$\varphi_0>$ 後面是第一個（上部）量子暫存器上的 n *個哈達瑪閘*。這給了新的狀態向量是

$$|\varphi_1\rangle = \frac{1}{\sqrt{2^n}}((\bigotimes_{k=1}^{n} |p_k^0\rangle + |p_k^1\rangle) \otimes ((\bigotimes_{y=1}^{L-1} |w_y^0\rangle) \otimes (|w_L{}^1\rangle)))$$
$$= \frac{1}{\sqrt{2^n}}(\sum_{P=0}^{2^n-1} |P\rangle \otimes ((\bigotimes_{y=1}^{L-1} |w_y^0\rangle) \otimes (|w_L{}^1\rangle)))。 \quad (5.32)$$

接下來，從圖 5.11，新的狀態向量| $\varphi_1>$ (5.32) 中的後面是一個量子閘 $|X^P$ 模 $N>=|X^{p_1\times 2^{n-1}+p_2\times 2^{n-2}+\cdots+p_n\times 2^{n-n}} \bmod N>$ 在兩個量子暫存器上運作。這給了新的狀態向量是

$$|\varphi_2\rangle = \frac{1}{\sqrt{2^n}}(\sum_{P=0}^{2^n-1} |P\rangle |X^P \bmod N\rangle)。 \quad (5.33)$$

因為順序為 X 模 N 為 r ，項為|(5.33) 中的$\varphi_2>$ 可以根據具有相同餘數的計算基礎狀態重新分組為 r *等價類* $XP^{模}N>$ 。第一個等效類別是 $\{r \times y + 0 \mid 0 \le y \le \lfloor (2^n - 0)/r \rfloor\}$ ，其中$\lfloor (2^n - 0)/r \rfloor$是取得小於等於$((2^n - 0)/r)$。第二個等效類別是 $\{r \times y + 1 \mid 0 \le y \le \lfloor (2^n - 1)/r \rfloor\}$.第三個等價類是 $\{r \times y + 2 \mid 0 \le y \le \lfloor (2^n - 2)/r \rfloor\}$.第四個等價類是 $\{r \times y + 3 \mid 0 \le y \le \lfloor (2^n - 3)/r \rfloor\}$.第 r 個等效類別是 $\{r \times y + (r-1) \mid 0 \le y \le \lfloor (2^n - (r-1))/r \rfloor\}$.並非所有等效類別都具有相同數量的元素。但是，如果 r 除 2^n ，則每個等效類別中的元素數量相同。我們假設對於 $0 \le$ *磷* $\le (r-1)$, $Y_P = \lfloor (2^n - P)/r \rfloor$。根據上面的陳述，我們重寫新的狀態向量| $\varphi_2>$ 在 (5.33) 中如下

$$|\varphi_2\rangle = \sum_{P=0}^{r-1} (\frac{1}{\sqrt{2^n}} \sum_{y=0}^{Y_P} |r \times y + P\rangle) |XP^{模}N\rangle 。 \quad (5.34)$$

為了方便演示，我們假設| $\varphi_{2P}>=$ $(\frac{1}{\sqrt{2^n}} \sum_{y=0}^{Y_P} |r \times y + P\rangle) 0 \le$ *磷* $\le (r-1)$。

如圖 5.11 所示，測量前的最後一步是在第一個（上部）量子暫存器上完成逆量子傅立葉變換（ **IQFT** ）。**疊加原理允許酉算符對每個**$|\varphi$進行一一操作。$_{2P}>$。因此，我們得到如下新的狀態向量

$$|\varphi_3> = \frac{1}{\sqrt{2^n}} \sum_{P=0}^{r-1} \sum_{y=0}^{Y_P} \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times(y\times r+P)} |我> |XP^{模}\mathrm{N}> \quad 。$$

$$= \sum_{i=0}^{2^n-1} \sum_{P=0}^{r-1} \sum_{y=0}^{Y_P} \frac{e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times(y\times r+P)}}{2^n} |我> |XP^{模}\mathrm{N}> \quad 。 \quad (5.35)$$

為了方便演示，我們假設φ0≤時 $_{iP}$ = ($\sum_{y=0}^{Y_P} \frac{e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times(y\times r+P)}}{2^n}$) *磷* ≤(r −1) 和 0≤ *我* ≤(^{2n} −1）。係數φ_{iP}和| $\varphi_{iP}|^2$ 隨後表示測量的幅度和機率|*我*> | $X^P \bmod N>$ 在圖 5.11 電路的輸出端。機率幅度可以相互抵消，同時增加測量合適狀態的機率。

為了方便表述，我們假設 $\boldsymbol{P}(i)$ 表示測量 | 的機率。*我*> | $X^P \bmod N>$ 在圖 5.11 電路的輸出端。基本機率論保證

$$\boldsymbol{P}(i) = \sum_{P=0}^{r-1} |\varphi_{iP}|^2 = \sum_{P=0}^{r-1} \left| \sum_{y=0}^{Y_P} \frac{e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times(y\times r+P)}}{2^n} \right|^2$$

$$= \sum_{P=0}^{r-1} |e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times P}|^2 \times \left| \sum_{y=0}^{Y_P} \frac{(e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times r})^y}{2^n} \right|^2 \quad (5.36)$$

由於基本機率論確保$|e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times P}|^2$= ($e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times P}$) ×($e^{\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times P}$) = 1，我們可以將(5.36) 中的 $\boldsymbol{P}(i)$ 改寫如下

$$\boldsymbol{P}(i) = \sum_{P=0}^{r-1} 1 \times \left| \sum_{y=0}^{Y_P} \frac{(e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times r})^y}{2^n} \right|^2$$

$$= \sum_{P=0}^{r-1} \left| \frac{1}{2^n} \sum_{y=0}^{Y_P} (e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times r})^y \right|^2$$

$$= \frac{1}{2^{2\times n}} \times \left(\sum_{P=0}^{r-1} \left| \sum_{y=0}^{Y_P} (e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times r})^y \right|^2 \right) 。 \quad (5.37)$$

為了實現幾何序列的求和，我們討論了*理想*情況和*實際*情況。若絕對值運算子的參數為$e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times r}$ = 1，則 ($i \times r / 2^n$) 是一個整數，我們可以將(5.37) 中的 $\boldsymbol{P}$ (i) 改寫如下

$$\boldsymbol{P}(i) = \frac{1}{2^{2\times n}} \times \left(\sum_{P=0}^{r-1} \left| \sum_{y=0}^{Y_P} 1^{y} \right|^2 \right) 。$$

$$= \frac{1}{2^{2\times n}} \times \left(\sum_{P=0}^{r-1} (Y_P + 1)^2 \right) 。 \quad (5.38)$$

我們稱（5.38）為*理想*情況。如果絕對值運算子的參數是$e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times r} \neq 1$，那麼（*我* $\times r / 2^{n}$) 不是整數，我們可以將(5.37) 中的 $\boldsymbol{P}(i)$ 改寫如下

$$\boldsymbol{P}(i) = \frac{1}{2^{2\times n}} \times \left(\sum_{P=0}^{r-1} \left| \frac{1-e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times r\times(1+Y_P)}}{1-e^{-\sqrt{-1}\times\frac{2\times\pi}{2^n}\times i\times r}} \right|^2 \right) 。 \quad (5.39)$$

我們將（5.39）稱為*實踐案例*。因為$|1- e^{\sqrt{-1}\times\theta}|^2 = 4\times\sin^2(\theta/2)$ 和 $\sin(-\theta/2) = -\sin(/2\ \theta)$ 和 $\sin^2(/2) - \theta = \sin(-\theta/2)\times\sin(-\theta/2) = (-\sin(\theta/2))\times(-\sin(\theta/2)) = \sin^2(\theta/2)$ ，我們可以將(5.39)中的 $\boldsymbol{P}(i)$ 改寫如下

$$\boldsymbol{P}(i) = \frac{1}{2^{2\times n}} \times \left(\sum_{P=0}^{r-1} \frac{4\times\sin^2(\frac{-2\times\pi\times i\times r\times(Y_P+1)}{2^n}\times\frac{1}{2})}{4\times\sin^2(\frac{-2\times\pi\times i\times r}{2^n}\times\frac{1}{2})} \right)$$

$$= \frac{1}{2^{2\times n}} \times \left(\sum_{P=0}^{r-1} \frac{\sin^2(\frac{\pi\times i\times r\times(Y_P+1)}{2^n})}{\sin^2(\frac{\pi\times i\times r}{2^n})} \right) 。 \quad (5.40)$$

5.14 因式分解的量子電路 15

我們要完成 $N = 15$ 的*質*因數分解。 $= 2$ 且 $N = 15$ 為 1（一）。這顯示 $X = 2$ 與 $N = 15$ *互質*。 命令 r 對 2 模 15 滿足 r 15. 由於表示≤$N = 15$ *的*位數有四位長，因此我們也只需要使用表示 r *值的四位*。如果 r 的值為偶數，則 $N = 15$ 的第一個非平凡因子等於 $\gcd(2^{r/2}+1, N)$， $N = 15$ 的第二個非平凡因子等於 $\gcd(2^{r/2}-1,\ N)$。

計算 這 命令 r 對 2 模 15 相當於計算週期 r 給定的神諭函數 O_f: { $p_1 p_2$ *第* 3 頁*第* 4 頁 | ∀ *壓力* ∈{0, 1} 為 $1 \le d \le 4$} →{$2^{p_1 p_2 p_3 p_4}$ (模 15) |∀ *壓力* ∈{0, 1} 為 $1 \le d \le 4$}。輸入變數 $p_1 p_2 p_3 p_4$ 是四位數。位 p_1 是*最高*有效位，位 p_4 是*最*

*低*有效位。對應的十進制值等於 $p_1 \times 2^{4-1} + p_2 \times 2^{4-2} + p_3 \times 2^{4-3} +$*第* $_4$ 頁 $\times 2^{4-4} = P$ 。 O_f的周期 r 滿足 $O_f(p_1 p_2 p_3 p_4) = O_f(p_1 p_2 p_3 p_4 + r)$ 到任兩個輸入 $(p_1 p_2 p_3 p_4)$ 和 $(p_1 p_2 p_3 p_4 + r)$。

為了實現模冪運算$2^{p_1 p_2 p_3 p_4}$（ mod 15），我們假設輔助變數 $w_1 w_2 w_3 w_4$是四位長。位 w_1是*最高*有效位，位 w_4是*最低*有效位。對應的十進制值等於 $w_1 \times 2^{4-1} + w_2 \times 2^{4-2} + w_3 \times 2^{4-3} + w_4 \times 2^{4-4} = W$ 。*前三位*中的每一位的初始值均為零(0)。*最低有效*位 w_4的初始值為一(1)。

X 模 *N* *階* *r* 的流程圖

圖 5.12 是計算 X 的階數 r 的流程圖 模 N 。在圖 5.12 中，在語句 S_1 中，它將輔助變數 W 的值設為 1。它將 X 的值設為 2，並將 N 的值設為 15。由於表示 N *的*位數為 4，因此將輔助變數 n 的值設定為 4。它設定索引

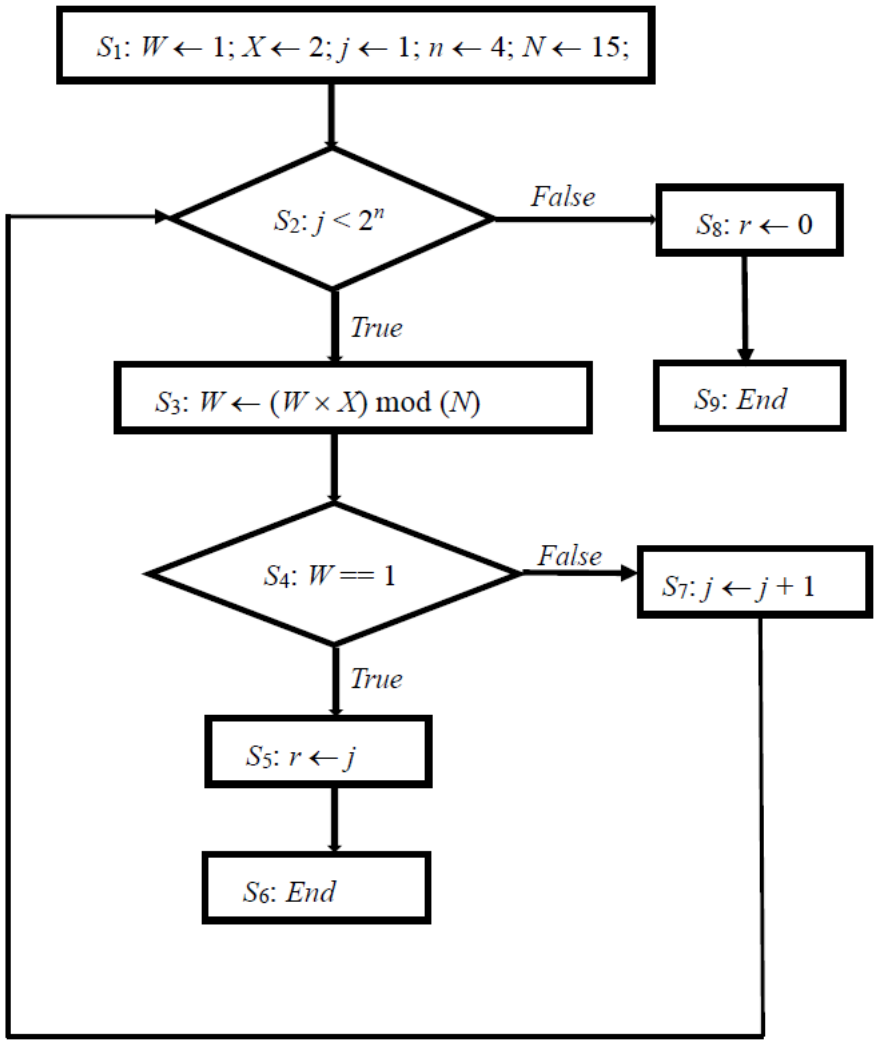


圖 5.12：流程圖 計算 X 模 N 的階 r 。

第一個迴圈的變數 j *為 1*。接下來，在語句 $S_{2中}$，執行第一個迴圈的條件判斷。如果 j 的值小於 2n ，則 *下一執行的*指令是語句 S_3 。否則，在語句 $S_{8中，它將階數}r$ 的值設為零。這就是說我們無法找到 X 模 N 的階 r 。接下來，在語句 $S_{9中}$，執行 *End 指令來終止求* X 模 N 的階 r *的*任務。

在語句 $S_{3中}$，完成一條乘法指令和一條取模指令，並將結果存入輔助變數 W。在第 j *次*執行語句 $S_{3時}$，它實際上完成了 $X^j \pmod N$ 並將結果儲存到輔助變數 W。*因為索引變數* j 的取值範圍是從 1 到 2^n-1，它實際上最多完成 $X^1 \pmod N$ 到$X^{2^n-1} \pmod N$。由於 $X^0 \pmod N = 1 \pmod N$，在語句 $S_{1中}$，它將輔助變數 W *設為* 1，以完成 $X^0 \pmod N$。

接下來，在語句 $S_{4中}$，執行條件判斷，判斷 W 的值是否等於 1。如果 W 的值等於 1，則下一執行的指令是語句 S_5 。在語句 $S_{5中，它將階數}r$ 的值設定為 j *的值*。這顯示我們已經找到了 X 模 N 的階 r 。接下來，在語句 $S_{6中}$，執行 *End* 指令來終止求 X 模 N 的階 r *的*任務。

語句 $S_{4中}$ W 的值不等於 1，則下一條執行的指令是語句 S_7 。接下來，在語句 $S_{7中，增加索引變數}j$ 的值。重複執行語句 S_2 到語句 S_7 ，直到語句 $S_{2中}$條件判斷變成*假*值或語句 $S_{6中}$執行 *End* 指令終止任務。由圖 5.12 可知，乘法指令和取模指令的總數最多為 (2^n-1)乘法指令和(2^n-1)模組化指令。這就是說，找到 X 模 N 的階 r *的成本*最多為完成(2^n-1)乘法指令和(2^n-1)模組化指令。

5.14。 2 模冪 $X^{P\,的實現}$ (模 N)

在圖 5.12 中，在順序模型中，它將每個模冪 $X^P \pmod N$ 計算為 0≤ *磷* ≤2n –1. 然而，在圖 5.11 中，在平行模型中，它同時計算每個模冪 $X^P \pmod N$ 到 0≤ *磷* ≤2n –1. 在圖 5.11 中，它使用了量子閘 $|X^P$ 模 $N> = |X^{p_1\times 2^{n-1}+p_2\times 2^{n-2}+\cdots+p_n\times 2^{n-n}} \bmod N>$ 在兩個量子暫存器上運作。計算模冪 $XP \pmod N$ *的方法*有兩個階段。第一階段利用模乘法透過對 $X \pmod N$ 求平方來計算 $X^2 \pmod N$。接下來，它透過對 $X^2 \pmod N$ *進行平方來計算* $X^4 \pmod N$。然後，它透過對 $X^4 \pmod N$ *求平方來*計算 $X^8 \pmod N$。透過這種方式，它繼續計算$X^{2^k} \pmod N$ 對於所有 k 直到 $(n-1)$。我們使用 $n = O(L)$，因此總共 $(n-1)$

= $O(L)$ 次平方運算的完成成本為 $O(L^2)$ 個（此成本假設用於平方運算的電路實現了一種乘法）。因此，第一階段的總成本為 $O(L^3)$。

方法的第二階段是完成以下觀察

$$X^P \pmod N = X^{p_1\times 2^{n-1} + p_2\times 2^{n-2} + \cdots + p_n\times 2^{n-n}} \text{ (模 } N\text{)}$$

$$= (X^{p_1\times 2^{n-1}} \text{ (模 } N\text{)}) \times_N (X^{p_2\times 2^{n-2}} \pmod N) \times_N \ldots \times_N (X^{p_n\times 2^{n-n}} \pmod N)。 \quad (5.41)$$

完成($n-1$) = $O(L)$ 模乘法，每次的成本為 $O(L^2)$，我們看到使用 $O(L^3)$ 閘可以計算 (5.41) 中的該乘積。這對於求 X 模 N 的階 r *來說是夠有效的*。當然，如果有更好的乘法電路，更有效的方法是可能的。

5.14.3 計算($X = 2$) 模 ($N = 15$)的階 r

($X = 2$) 模數 ($N = 15$) *的階數*為最小*正*整數 r ，使得 $2^r = 1$（模 15）。因為 $r \le (N = 15)$*和表示*$(N = 15)$的位數是四位長，所以表示 r 的位數是四位長。因此，n 個輸入變數 $p_1 p_2 p_3 p_4$ 是四位數。位 p_1 是*最高有效位*，位 p_4 是*最低有效位*。對應的十進制值等於 $p_1\times 2^{4-1} + p_2\times 2^{4-2} + p_3\times 2^{4-3} +$*第* $_4$ *頁* $\times 2^{4-4} = P$ 。也就是說 P *的取值範圍*是從 0 到 15。如果 P 是滿足下列條件的*最小*正整數 $2^{P = 1 \pmod{15}}$，*則* r 的值等於 P *的值*。我們用 $p_d{}^{0}$ *來表示* p_d 的值為 0 時為 $1 \le d \le 4$ 並套用 $p_d{}^{1}$ 表示 p_d *的值為* 1 for $1 \le d \le 4$.

$^P \pmod{15}$≤的餘數為 0 *磷* ≤15 是從零到十四，我們假設輔助變數 $w_1 w_2 w_3 w_4$ 是四位長，我們用它來儲存 $2^P \pmod{15}$ 到 0 的≤計算結果 *磷* ≤15.我們用 $w_d{}^{0}$ *來表示* w_d 的值為 1 時為零≤ $d \le 4$ 並應用 $w_d{}^{1}$ 表示 w_d *的值為* 1 為 $1 \le d \le 4$.位 w_1 是*最高有效位*，位 w_4 是*最低有效位*。對應的十進制值等於 $w_1\times 2^{4-1} + w_2\times 2^{4-2} + w_3\times 2^{4-3} + w_4\times 2^{4-4} = W$ 。*前三位*中每一位的初始值為 0。*最低有效位* w_4 的初始值為 1。這就是說 $W = w_1{}^{0}\times 2^{4-1} + w_2{}^{0}\times 2^{4-2} + w_3{}^{0}\times 2^{4-3} + w_4{}^{1}\times 2^{4-4} = 1$。

計算 這 命令 r 對 2 模 15 相當於計算週期 r 給定的神諭函數 O_f: { $p_1 p_2$ *第* $_3$ *頁第* $_4$ *頁* | ∀ *壓力* ∈{0, 1} 為 $1 \le d \le 4$} →{$2^{p_1 p_2 p_3 p_4}$ (模 15) |∀ *壓力* ∈{0, 1} 為 $1 \le d \le 4$}。 O_f的周期 r 滿足 $O_f(p_1 p_2 p_3 p_4) = O_f(p_1 p_2 p_3 p_4 + r)$ 到任兩個輸入 ($p_1 p_2 p_3 p_4$) 和 ($p_1 p_2 p_3 p_4 + r$)。第一階段計算 $O_f(p_1 p_2 p_3 p_4) = 2^{p_1 p_2 p_3 p_4}$ (mod 15) 是使用模乘法透過對 2 (mod 15) 求平方來計算 $2^2 \pmod{15}$。我們得到以下結果

2^{2}(模 15) = (2 (模 15))^{2} = (2 (模 15)) $\times_{15}$ (2 (模 15)) = 4 (模 15) = 4。

接下來，它透過對 2^{2}(mod 15) 進行平方來計算 2^{4}(mod 15)，我們得到以下結果

2^{4}(模 15) = (2^{2}(模 15))^{2} = (2^{2}(模 15)) $\times_{15}$ (2^{2}(模 15)) = 4^{2}(模 15) = 1。

然後，透過對 2^{4}(mod 15) 求平方來計算 2^{8}(mod 15)，我們得到以下結果

$$2^{8}(\text{mod } 15) = (2^{4}(\text{mod } 15))^{2} = (2^{4\,(\text{mod } 15)}) \times_{15} (2^{4}(\text{mod } 15)) = 1^{2}(\text{mod } 15) = 1. \quad (5.44)$$

接下來，第二階段計算 $O_f(p_1 p_2 p_3 p_4) = 2^{p_1 p_2 p_3 p_4}$ （ mod 15）是完成以下觀察

$$2^{P}(\text{模 } 15) = 2^{p_1\times 2^3 + p_2\times 2^2 + p_3\times 2^1 + p_4\times 2^0}(\text{模 } 15) = (2^{p_1\times 2^3}(\text{模 } 15)) \times_{15} (2^{p_2\times 2^2}(\text{模 } 15)) \times_{15} (2^{p_3\times 2^1}(\text{模 } 15)) \times_{15} (2^{p_4\times 2^0}(\text{模 } 15))。 \quad (5.45)$$

p_1的值等於一 (1)，則 $(2^{p_1{}^1\times 2^3} \pmod{15}) = (2^{8}(\text{mod } 15)) = 1$。$2^{p_1{}^0\times 2^3} \pmod{15})$ $= (2^{0}(\text{mod } 15)) = 1.$ 這就是說 $(2^{p_1\times 2^3} (\text{mod}_{15}))) = 1$ $2^{p_2{}^1\times 2^2}$。 $(\text{mod } 15)) = (2^{4}$ $(\text{mod } 15)) = 1$。$2^{p_2{}^0\times 2^2} \pmod{15})) = (2^{0}(\text{mod } 15)) = 1.$ 這表示 $(2^{p_2\times 2^2} \pmod{15})) = 1$。

$$2^{P}(\text{mod } 15) = (2^{p_3\times 2^1}(\text{mod } 15)) \times_{15} (2^{p_4\times 2^0}(\text{mod } 15))。 \quad (5.46)$$

根據 (5.46) 中的方程， O_f的 16 個輸出採用 $p_1{}^0 p_2{}^{0\,\text{的每個輸入}} p_3{}^0 p_4{}^0$ 至 $p_1{}^1 p_2{}^1 p_3{}^1 p_4{}^1$ 隨後為 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4 和 8。 2)、四(4)和八(8)依序為 $w_1{}^0 w_2{}^0 w_3{}^0 w_4{}^1$, $w_1{}^0 w_2{}^0 w_3{}^1 w_4{}^0$, $w_1{}^0 w_2{}^1 w_3{}^0 w_4{}^0$和 $w_1{}^1 w_2{}^0 w_3{}^0 w_4{}^0$ 。 O_f的頻率 f 等於每十六個輸出的週期數。這給了 $r \times f = 16$ 。1這意味著每十六個輸出的周期數為*四且* O_f的頻率 f 等於四。 O_f的周期 r 是 O_f的頻率 f *的倒數*。因此，我們得到 $r = \frac{1}{\frac{f}{16}} = \frac{16}{f} = \frac{16}{4} = 4$ 和 $r \times f = \times 4\ 4 = 16$ 。

5.14.4 減少模冪運算 2^P （模 15）

在(5.46) 中，神諭函數為 $O_f(p_1 p_2 p_3 p_4) = 2^{p_1 p_2 p_3 p_4} \pmod{15} = (2^{p_3 \times 2^1} \pmod{15}) \times_{15} (2^{p_4 \times 2^0} \pmod{15})$。若位 p_3 的值等於一 (1)，則 $(2^{p_3{}^1 \times 2^1} \pmod{15}) = (2^2 \pmod{15}) = 4$. 否則，$(2^{p_3{}^0 \times 2^1} \pmod{15}) = (2^0 \pmod{15}) = 1$。 這就是說，如果位 p_3 的值等於一 (1)，則指令「$(2^{p_3{}^1 \times 2^1} \pmod{15}) = (2^2 \pmod{15}) = 4$” 可以透過乘以任意輔助變數 $w_1{}^0 w_2{}^0$ 來實現 $w_3{}^0 w_4{}^1 \times 2^2$ 。否則，指令「$(2^{p_3{}^0 \times 2^1} \pmod{15}) = (2^0 \pmod{15}) = 1$” 未實現。

類似地，如果位 p_4 的值等於一 (1)，則 $(2^{p_4{}^1 \times 2^0} \pmod{15}) = (2^1 \pmod{15}) = 2$。$2^{p_4{}^0 \times 2^0} \pmod{15}) = (2^0 \pmod{15}) = 1$ $2^{p_4{}^1 \times 2^0}$。 $\pmod{15}) = (2^1 \pmod{15}) = 2$” 可以乘以輔助變數 $w_1{}^0 w_2{}^0$ 來實現 $w_3{}^0 w_4{}^1 \times 2^1$ 。否則，指令「$(2^{p_4{}^0 \times 2^0} \pmod{15}) = (2^0 \pmod{15}) = 1$” 不執行。

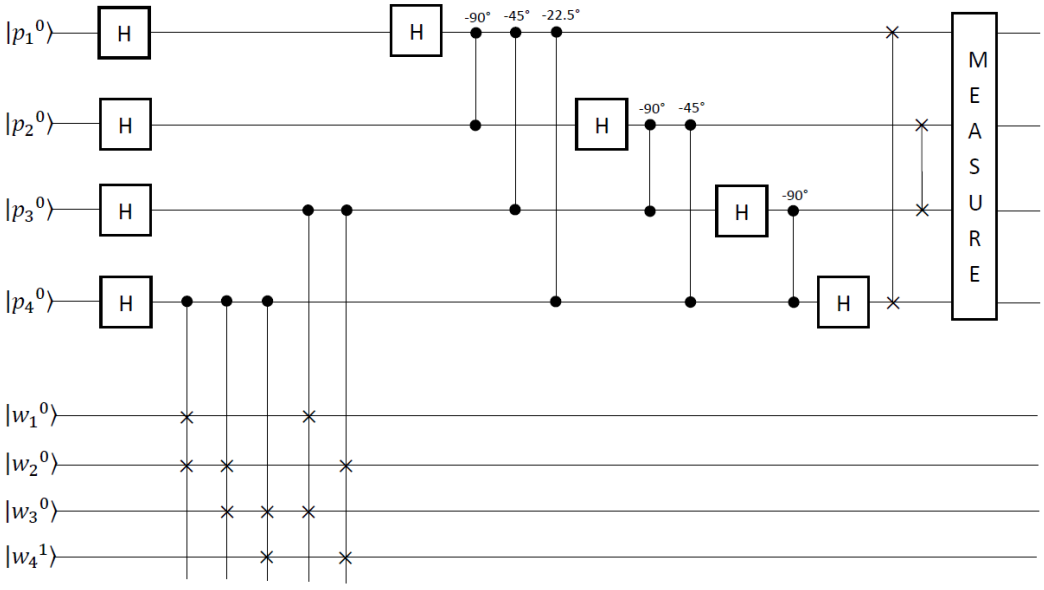
當然，使用簡單的位移位可以實現對任何二進位輔助變數乘以 2（或實際上是 2 的任意冪）。計算 2^P 需要對任何二進位輔助變數進行 *P 乘以 2。完成 P 次*左移即可對任意二進位輔助變數實現 *P* 乘 2。這意味著計算 2^P 需要對任何二進位輔助變數完成 *P 次*左移。

例如，在我們的範例中，我們使用位元 p_3 和 p_4 作為受控位元。如果受控位元 p_4 的值等於一 (1)，則我們對二進位輔助變數 $w_1{}^0 w_2{}^0$ 使用一次左移位 $w_3{}^0 w_4{}^1$ 執行指令「$(2^{p_4{}^1 \times 2^0} \pmod{15}) = (2^1 \pmod{15}) = 2$”。左移一次就是將每一位 w_k 換成 $1 \le k \le 4$ 具有下一個最高權重位置。第一次執行時，將位元 $w_1{}^0$ 與位元 $w_2{}^0$ 交換，結果為 $w_2{}^0 w_1{}^0 w_3{}^0 w_4{}^1$ 。接下來，在第二次執行時，將位元 $w_1{}^0$ 與位元 $w_3{}^0$ 交換，結果為 $w_2{}^0 w_3{}^0 w_1{}^0 w_4{}^1$ 。接下來，在第三次執行時，將位元 $w_1{}^0$ 與位元 $w_4{}^1$ 交換，結果為 $w_2{}^0 w_3{}^0 w_4{}^1 w_1{}^0$ 。

$w_2{}^0 w_3{}^0$ 對應的十進位值 $w_4{}^1 w_1{}^0$ 是二 (2)。這意味著它執行指令“$(2^{p_4{}^1\times 2^0} \pmod{15}) = (2^1 \pmod{15}) = 2$”。我們可以使用三個 ***CSWAP*** 實施這些措施的大門。類似地，如果受控位 p_3 的值等於一（1），那麼我們完成兩位移位，以實現指令「$(2^{p_3{}^1\times 2^1} \pmod{15}) = (2^2 \pmod{15}) = 4$”。移位兩位是指將加權位置$(2^3)$的位元與加權位置$(2^1)$的另一位元交換，並將加權位置$(2^2)$與加權位置$(2^2)$的另一位元交換。).我們可以利用兩個 ***CSWAP*** 閘來實現它們。這種減少使得我們不必實現乘法電路和模組化電路。

5.14.5 初始化量子電路的量子暫存器以求$(X = 2)$ 模 $(N = 15)$的階次 *r*

我們使用圖 5.13 的量子電路來求$(X = 2)$ 模 $(N = 15)$的階數 r 。第一個（上部）量子暫存器有四個量子位元。位元$|p_1\rangle$ 是*最高有效位元*和位元 $|p_4\rangle$ 是*最低有效位元*。對應的十進制值等於 $p_1\times 2^{4-1} + p_2\times 2^{4-2} + p_3\times 2^{4-3} + p_4\times 2^{4-4} = P$ 。每個的初始狀態

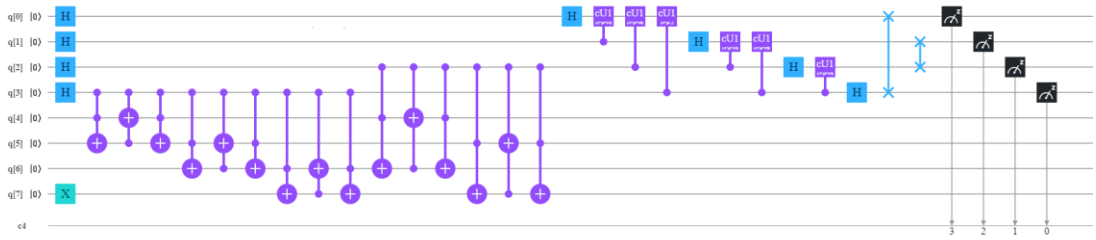


$(X = 2)$ 模 $(N = 15)$的階數 r *的量子電路*。

量子比特$|p_d\rangle$ 為 $1 \le d \le 4$ 設定為狀態$|0\rangle$。第二個（下部）量子暫存器有四個量子位元。位元$|w_1\rangle$ 是*最高有效位元*和位元$|w_4\rangle$ 是*最低有效位元*。對應的十進制值等於 $w_1\times 2^{4-1} + w_2\times 2^{4-2} + w_3\times 2^{4-3} + w_4\times 2^{4-4} = W$ 。前三個量子位元的初始狀態 $|w_d\rangle$ 為 $1 \le d \le 3$ 設定為狀態$|0\rangle$。最低有效量子位元的初始狀態 $|w_d$

> 為 $1 \le d \le 3$ 設定為狀態|1>。

IBM 量子電腦中具有 *32 個*量子位元的 Open QASM *模擬器的*後端。程式是求($X = 2$) 模 ($N = 15$)的階數 r 。圖 5.14 是清單 5.2 中程式對應的量子電路，是實現圖 5.13 中求($X = 2$)模($N = 15$)的階 r *的量子電路*。



($X = 2$) 模數 ($N = 15$)階次 r *的*量子電路。

聲明“OPENQASM 2.0；”清單 5.2 的第一行表示程式是用 Open QASM 2.0 版本編寫的。然後，語句“ include”qelib1.inc”； 」 清單 5.2 的第二行是繼續解析檔案「 q elib1.inc」 ，就好像該檔案的內容被貼到 include 語句的位置，其中檔案「 q elib1.inc 」是 **Quantum Experience (QE) 標準標頭**，且路徑是相對於目前工作指定的 目錄。

```
1. 開放 QASM 2.0；
2. 包括“qelib1.inc”；
3. qreg q[8];
4. 克雷格 c[4]；
5. xq[7];
```

($X = 2$) 模 ($N = 15$)的階 r 的程序。

接下來，語句「qreg q[8] ;清單 5.2 的第三行是聲明程式中有 8 *個*量子位元。在圖 5.14 的左上角，八個量子位元依序為 q[0]、q[1]、q[2]、q[3]、q[4]、q[5]、q[6]和 q[7]。每個量子位元的初始值被設定為狀態|0>。我們使用四個量子比特 q[0]、q[1]、q[2]和 q[3]分別編碼四個量子位元| p_1 >, | p_2 >, | p_3 > 和 | p_4 > 圖 5.13。我們應用四個量子比特 q[4]、q[5]、q[6]和 q[7]分別編碼四個量子位元| w_1 >, | w_2 >, | w_3 > 和 | w_4 > 在圖 5.13 中。

為了方便我們解釋，q[k] ^{0} 代表 $0 \le k \le 7$ 是表示 q[k]的值為 0，q[k] ^{1} 為 $0 \le k \le 7$ 表示 q[k]的值 1。接下來，語句“ creg c[4];”清單 5.2 的第四行是聲明程序中有四個經典位。在圖 5.14 的左下角，四個經典位依序為 c[0]、c[1]、c[2]和 c[3]。每個經典位元的初始值設定為零 (0)。為了方便我們解釋，c[k] ^{0} 代表 $0 \le k \le 3$ 是表

示 c[k]的值 0，c[k] ^{1} 表示 $0 \le k \le 3$ 表示 c[k]的值 1。四個初始經典位元 c[3] ^{0} c[2] ^{0} c[1] ^{0} c[0] ^{0} 對應的十進位值為 $2^{3} \times c[3]^{0} + 2^{2} \times c[2]^{0} + 2^{1} \times c[1]^{0} + 2^{0} \times c[0]^{0}$ 。這顯示經典位 c[3] ^{0} 是最高有效位，經典位 c[0] ^{0} 是最低有效位。接下來，語句「xq[7];清單 5.2 第五行的「」是將量子位元 |q[7]> 的狀態 |0> 轉換為狀態 |1>。為了方便解釋，求($X = 2$) 模 ($N = 15$)的階數 r *的初始狀態向量為*

$$|\Omega_0\rangle = |q[0]^{0}\rangle |q[1]^{0}\rangle |q[2]^{0}\rangle |q[3]^{0}\rangle |q[4]^{0}\rangle |q[5]^{0}\rangle |q[6]^{0}\rangle |q[7]^{1}\rangle 。 \quad (5.47)$$

5.14.6 計算 $2^{P} \pmod{15}$的量子疊加

在初始狀態向量$|\Omega_0\rangle$ 在 (5.47) 中，量子位元 |q[0] ^{0} > |q[1] ^{0} > |q[2] ^{0} > |q[3] ^{0} > 編碼四個量子位元$|p_1^{0}\rangle, |p_2^{0}\rangle, |p_3^{0}\rangle$ 和 |圖 5.13 中第一個暫存器的 p_4^{0} >和 是*精度*暫存器。我們使用精度暫存器來表示 P 的值，我們將其傳遞給模冪 $2^{P} \pmod{15}$。我們將利用量子疊加來並行計算多個 P 值的模冪 $2^{P \pmod{15}}$ ，因此我們使用四個語句「hq[0];」、「hq[1];」、「hq[2」];"和"總部[3]；"清單 5.2 中的*第六*行到*第九*行將*精度*暫存器放入

清單 5.2 繼續…

6. 總部[0]；
7. 總部[1]；
8. 總部[2]；
9. 總部[3]；

所有可能值的疊加。因此，我們有以下新的狀態向量

$$|\Omega_1\rangle = \left(\frac{1}{\sqrt{2}}(|q[0]^{0}\rangle + |q[0]^{1}\rangle)\right)\left(\frac{1}{\sqrt{2}}(|q[1]^{0}\rangle + |q[1]^{1}\rangle)\right)\left(\frac{1}{\sqrt{2}}(|q[2]^{0}\rangle + |q[2]^{1}\rangle)\right)$$

$$\left(\frac{1}{\sqrt{2}}(|q[3]^{0}\rangle + |q[3]^{1}\rangle)\right)(|q[4]^{0}\rangle |q[5]^{0}\rangle |q[6]^{0}\rangle |q[7]^{1}\rangle)$$

$$= \frac{1}{\sqrt{2^4}}\left(\sum_{P=0}^{2^4-1}|P\rangle\right)(|q[4]^{0}\rangle |q[5]^{0}\rangle |q[6]^{0}\rangle |q[7]^{1}\rangle)。 \quad (5.48)$$

這樣使得每個狀態 $(|P\rangle |q[4]^{0}\rangle |q[5]^{0}\rangle |q[6]^{0}\rangle |q[7]^{1}\rangle)$ (5.48) 中的新狀態向量 $(|\Omega_1\rangle)$ 準備就緒被視為並行計算的單獨輸入。

5.14.7 實現條件乘以 2 計算 $2^P \pmod{15}$

在初始狀態向量$|\Omega_0>$ 在 (5.47) 中，量子位元 $|q[4]^0>$ $|q[5]^0>$ $|q[6]^0>$ $|q[7]^1>$ 編碼四個量子位元$|w_1^0>, |w_2^0>, |w_3^0>$ 和 |圖 5.13 中第二個暫存器的$w_4^1>$和 是*輔助*暫存器。我們現在希望在精度暫存器內的輸入疊加上完成 (5.46) 中的模冪 $2^P \pmod{15} = (2^{p_3\times 2^1} \pmod{15}) \times_{15} (2^{p_4\times 2^0} \pmod{15})$ ，我們將應用輔助暫存器來保存和儲存結果。我們使用三個 ***CSWAP 閘來實現***條件乘以 2 的指令 $(\pmod{15})$。$2^{p_4\times 2^0}$三個語句"ccx q[3],q[4],q[5];","ccx q[3],q[5],q[4];"和"ccx q[3]，q[4]，q[5]；"清單 5.2 中從第 10 行到第 12 行是三個 ***CCNOT*** 閘，用於實現*第一個* ***CSWAP*** 閘如圖 5.13 所示。在第一個 ***CSWAP*** 閘中，量子位 q[3]是其受控位，量子位 q[4]和 q[5]是其目標位。在 ***CCNOT*** 指令中，第一操作數和第二操作數是其兩個控制位，第三操作數是其目標位。

清單 5.2 繼續…

```
//實作第一個 CSWAP 閘。
10. ccx q[3],q[4],q[5];
11. ccx q[3],q[5],q[4];
12. ccx q[3],q[4],q[5];

//實作第二個 CSWAP 閘。
13. ccx q[3],q[5],q[6];
14. ccx q[3],q[6],q[5];
15. ccx q[3],q[5],q[6];

//實作第三個 CSWAP 閘。
16. ccx q[3],q[6],q[7];
17. ccx q[3],q[7],q[6];
18. ccx q[3],q[6],q[7];
```

三個 ***CCNOT 閘***將工作暫存器的加權位置(2^3)處的量子位元與工作暫存器的加權位置(2^2)處的量子位元交換。這意味著新的狀態向量是

$$|\Omega_2> = \frac{1}{\sqrt{2^4}} (|q[0]^0> + |q[0]^1>)(|q[1]^0> + |q[1]^1>)(|q[2]^0> + |q[2]^1>)(|q[3]^0> |q[4]^0> |q[5]^0> |q[6]^0> |q[7]^1> + |q[3]^1> |q[5]^0> |q[4]^0> |q[6]^0> |q[7]^1>)。 \quad (5.49)$$

接下來，在圖 5.13 的第二個 ***CSWAP*** 閘中，量子位 q[3] 是其受控位，量子位 q[5] 和 q[6] 是其目標位。三個 ***CCNOT*** 閘“ ccx q[3],q[5],q[6] ;”,“ ccx q[3],q[6],q[5] ;”和“ ccx q[3],q[5],q[6];清單 5.2 中的第 13 行到第 15 行是實現*第二個* ***CSWAP*** 閘如圖 5.13 所示。這表示新的狀態向量是

$$|\Omega_3> = \frac{1}{\sqrt{2^4}} (|q[0]^0> + |q[0]^1>)(|q[1]^0> + |q[1]^1>)(|q[2]^0> + |q[2]^1>)(|q[3]^0> |q[4]^0> |q[5]^0> |q[6]^0> |q[7]^1> + |q[3]^1> |q[5]^0> |q[6]^0> |q[4]^0> |q[7]^1>)。 \quad (5.50)$$

那麼，在圖 5.13 的第三個 ***CSWAP*** 閘中，量子位 q[3]是其受控位，量子位 q[6]和 q[7]是其目標位。三個 ***CCNOT*** 閘“ ccx q[3],q[6],q[7] ;”，“ ccx q[3],q[7],q[6] ;”和“ ccx q[3],q[6],q[7] ;”清單 5.2 中的第 16 行到第 18 行是實現*第三個* ***CSWAP*** 閘如圖 5.13 所示。這意味著新的狀態向量是

$$|\Omega_4> = \frac{1}{\sqrt{2^4}} (|q[0]^0> + |q[0]^1>)(|q[1]^0> + |q[1]^1>)(|q[2]^0> + |q[2]^1>)(|q[3]^0> |q[4]^0> |q[5]^0> |q[6]^0> |q[7]^1> + |q[3]^1> |q[5]^0> |q[6]^0> |q[7]^1> |q[4]^0>)。 \quad (5.51)$$

在新的狀態向量|Ω_4>在(5.51)中，若精度暫存器中的最低有效量子位元 q[3]為|1>狀態，則其工作暫存器的值為二(2)。否則，其工作暫存器的值不會改變並且仍然是一(1) 。

5.14.8 實現條件乘以 4 來計算 $2^P \pmod{15}$

因為工作暫存器在新的狀態向量| Ω_4 >(5.51)中儲存並儲存了計算結果($2^{p_4 \times 2^0} \pmod{15}$)，接下來我們要完成指令($2^{p_3 \times 2^1} \pmod{15}$)。如果編碼量子位元|的量子位元|q[2]>的值 p_3 > 在精度暫存器中是一 (1)，則表示完成指令($2^{p_3 \times 2^1} \pmod{15}$) 需要在工作暫存器上再進行兩次乘以二 (2)。我們使用兩個 ***CSWAP*** *閘來實現*條件乘以 4 的指令 ((mod 15))。$2^{p_3 \times 2^1}$在圖 5.13 中，第四個 ***CSWAP*** *閘的作用是將*工作暫存器的加權位置 (2^3)處的量子位元與工作暫存器

的加權位置 (2^1) 處的另一個量子位元進行交換，如果該量子位的值為精度暫存器的加權位置(2^1)是狀態|1>。接下來，在圖 5.13 中，第五個 ***CSWAP 閘***將工作暫存器的加權位置 (2^2) 處的量子位元與工作暫存器的加權位置 (2^0) 處的另一個量子位元交換，如果量子的值精度暫存器的加權位置(2^1)處的位元是狀態|1>。

三個語句“ccx q[2],q[4],q[6];”,“ccx q[2],q[6],q[4];”和“ccx q[2],q[4],q[6];”清單 5.2 中從第 19 行到第 21 行是實現*第四個的三個* ***CCNOT 閘*** ***CSWAP*** 閘如圖 5.13 所示。在第四個 ***CSWAP*** 閘中，量子位 q[2]是其受控位，量子位 q[4]和 q[6]是其目標位。在 ***CCNOT*** 指令中，第一操作數和第二操作數是其兩個控制位，第三操作數是其目標位。

清單 5.2 繼續…

```
//執行第四個 CSWAP 門。
19. ccx q[2],q[4],q[6];
20. ccx q[2],q[6],q[4];
21. ccx q[2],q[4],q[6];

//執行第五個 CSWAP 門。
22. ccx q[2],q[5],q[7];
23. ccx q[2],q[7],q[5];
24. ccx q[2],q[5],q[7];
```

三個 ***CCNOT 閘***將工作暫存器的加權位置(2^3)處的量子位元與工作暫存器的加權位置(2^1)處的量子位元交換。這意味著新的狀態向量是

$$|\Omega_5> = \frac{1}{\sqrt{2^4}} (|q[0]^0> + |q[0]^1>)(|q[1]^0> + |q[1]^1>)(|q[2]^0>|q[3]^0>|q[4]^0>|q[5]^0>|q[6]^0>$$
$$|q[7]^1> + |q[2]^0>|q[3]^1>|q[5]^0>|q[6]^0>|q[7]^1>|q[4]^0> + (|q[2]^1>|q[3]^0>|q[6]^0>|q[5]^0>$$
$$|q[4]^0>|q[7]^1> + |q[2]^1>|q[3]^1>|q[7]^1>|q[6]^0>|q[5]^0>|q[4]^0>)。 \quad (5.52)$$

接下來，在圖 5.13 的第五個 ***CSWAP*** 閘中，量子位 q[2] 是其受控位，量子位 q[5] 和 q[7] 是其目標位。三個 ***CCNOT*** 閘“ ccx q[2],q[5],q[7] ;”，“ ccx q[2],q[7],q[5] ;”和“ ccx q[2],q[5],q[7] ;”清單 5.2 中的第 22 行到第 24 行是實現*第五個* ***CSWAP*** 閘如圖 5.13 所示。這意味著新的狀態向量是

$|\Omega_6> = \frac{1}{\sqrt{2^4}}$ (|q[0] ^{0}> + |q[0] ^{1}>) (|q[1] ^{0}> + |q[1] ^{1}>) (|q[2] ^{0}> |q[3] ^{0}> | q[4] ^{0}> |q[5] ^{0}> |q[6] ^{0}> |q[7] ^{1}> + |q[2] ^{0}> |q[3] ^{1}> |q[5] ^{0}> |q[6] ^{0}> |q[7] ^{1}> |q[4] ^{0}> + (|q[2] ^{1}> |q[3] ^{0}> |q[6] ^{0}> |q[7] ^{1}> |q[4] ^{0}> |q[5] ^{0}> + |q[2] ^{1}> |q[3] ^{1}> |q[7] ^{1}> |q[4] ^{0}> |q[5] ^{0}> |q[6] ^{0}>)。 (5.53)

在新的狀態向量$|\Omega_6>$在 (5.53) 中，它顯示了我們現在如何設法計算 (5.46) 中的 $2^P \pmod{15} = (2^{p_3 \times 2^1} \pmod{15}) \times_{15} (2^{p_4 \times 2^0} \pmod{15})$ 中精度寄存器中 *P 的每個值*疊加。

5.14.9 實現四個量子位元的量子傅立葉逆變換

在圖 5.13 中，透過對精度暫存器完成逆量子傅立葉變換，它有效地將精度暫存器狀態轉換為*週期*訊號分量頻率的疊加。十二條聲明來自

清單 5.2 繼續…

```
// 實作量子傅立葉逆變換。
25. 總部[0]；
26. cu1(-2*pi*1/4) q[1],q[0];
27. cu1(-2*pi*1/8) q[2],q[0];
28. cu1(-2*pi*1/16) q[3],q[0];

29. 總部[1]；
30. cu1(-2*pi*1/4) q[2],q[1];
31. cu1(-2*pi*1/8) q[3],q[1];

32. 總部[2]；
33. cu1(-2*pi*1/4) q[3],q[2];

34. 總部[3]；
35. 交換 q[0],q[3]；
36. 交換 q[1],q[2]；
```

的第 25 行到第 36 行在精度暫存器上實現了逆量子傅立葉變換。他們採用新的

狀態向量| (5.53) 中的$\Omega_6>$ 作為其輸入狀態向量。他們產生以下新的狀態向量

$$|\Omega_7> = (\frac{1}{\sqrt{2^2}} (|q[0]^0> |q[1]^0> |q[2]^0> |q[3]^0>) + \frac{1}{\sqrt{2^2}} (|q[0]^0> |q[1]^1> |q[2]^0> |q[3]^0>)$$

$$+ \frac{1}{\sqrt{2^2}} (|q[0]^1> |q[1]^0> |q[2]^0> |q[3]^0>) + \frac{1}{\sqrt{2^2}} (|q[0]^1> |q[1]^1> |q[2]^0> |q[3]^0>))$$

$$(|q[4]^0> |q [5]^0> |q[6] 0 > |q[7]^{1>} + | q[5]^0> |q[6]^0> |q[7]^1> |q[4]^0> + |q[6]^0> |q[7]^1>$$

$$|q[4]^0> |q[5]^0> + |q[7]^1> |q[4]^0> |q[5]^0> |q[6]^0>) 。 (5.54)$$

5.14.10 讀取量子結果

最後，四個語句“measure q[0] -> c[3];”、“measure q[1] -> c[2];”、“measure q[2] -> c[1];”和“測量 q[3] -> c[0]；”清單 5.2 中的第 37 行到第 40 行實現了圖 5.13 中精度暫存器的測量。它們測量精度暫存器的四個量子位元 q[0]、q[1]、q[2] 和 q[3]。它們透過涵蓋四個經典位 c[3]、c[2]、c[1] 和 c[0] 來記錄測量結果。

清單 5.2 繼續…

```
// 對精度暫存器進行一次測量
37. 測量 q[0] -> c[3]；
38. 測量 q[1] -> c[2]；
39. 測量 q[2] -> c[1]；
40. 測量 q[3] -> c[0]；
```

IBM 量子電腦的 32 個量子位元的後端*模擬器中*，我們使用「run」指令來執行清單 5.2 中的程式。測量結果如圖 5.15 所示。從圖 5.15 中，我們得到計算基礎狀態 0000 (c[3] = 0 = q[0] = |0>, c[2] = 0 = q[1] = |0>, c[1] = 0 = q[2] = |0> 和 c[0] = 0 = q[3] = |0>) 的機率為 24.512% (0.24512)。另一方面，我們得到計算基礎狀態 0100 (c[3] = 0 = q[0] = |0>, c[2] = 1 = q[1] = |1>, c[1] = 0 = q[2] = |0> 和 c[0] = 0 = q[3] = |0>) 的機率為 23.145% (0.23145)。或者，我們得到計算基礎狀態 1000 (c[3] = 1 = q[0] = |1>, c[2] = 0 = q[1] = |0>, c[1] = 0 = q[2] = |0> 和 c[0] = 0 = q[3] = |0>) 的機率為 27.148% (0.27148)。另一方面，我們得到計算基礎狀態 1100 (c[3] = 1 = q[0] = |1>, c[2] = 1 = q[1] = |1>, c[1] = 0 = q[2] = |0> 和 c[0] = 0 = q[3] = |0>) 的機

率為 25.195% (0.25195)。

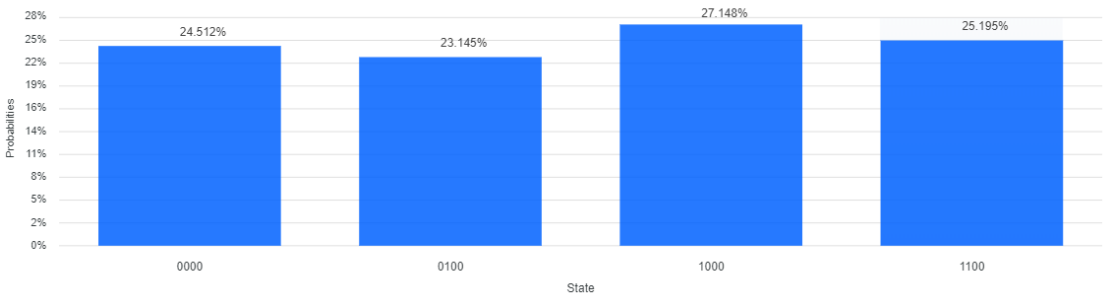


圖 5.15：計算基礎狀態 0000 的機率為 24.512% (0.24512)，計算基礎狀態 0100 的機率為 23.145% (0.23145)，計算基礎狀態 1000 的機率為 27.148% (0.271. % (0.25195)。

我們選擇計算基礎狀態 0100 (c[3] = 0 = q[0] = |0>, c[2] = 1 = q[1] = |1>, c[1] = 0 = q[2] = |0> 和 c[0] = 0 = q[3] = |0>)，測量結果為機率 23.145% (0.23145)。因為計算基礎狀態 1000 的十進制值為四(4)且($2^4/4$)是有理數，所以我們使用圖 5.2 中的連分數 al 演算法 如果 $c = 2^4 = 16$ 且 $d = 4$ 以及相應的收斂，則確定 (c/d) *的連續分數表示*。從第一次執行語句 S_0 到語句 S_2 ，得到 $i = 1$、 $q[1] = c/d = 16/4 = 4$ 和 $r = 16 \pmod 4 = 0$。) 轉換其整數和小數部分且不反轉其小數部分，

$$\frac{16}{4} = 4 + \frac{0}{4} = 4 \text{ 。} \qquad (5.55)$$

r 的值等於 0，因此從第一次執行語句 $S_{3\text{開始}}$，它會傳回 *true* 。因此，接下來，從第一次執行語句 $S_{4\text{開始}}$，答案是 (16 / 4) 的連分數表示

$$\frac{16}{4} = (q[1] = 4) = 4. \quad (5.56)$$

接下來，從第一次執行語句 $S_{5\text{開始}}$，終止連分數 a 演算法的執行。對於有理數 (16 / 4)，第一個收斂是 $(q[1]) = 4 = \frac{4}{1}$且最接近( ) $\frac{16}{4}$，且分子小於 15 。因此，我們檢查 $2^4 \pmod{15}$ 等於一 (1) ，我們發現階數 r 為四 (4)。因為階數 r 是偶數，根據**引理 5-2** ，我們使用歐幾里德演算法來計算 $\gcd(15, 2^{\frac{4}{2}} + 1)$ 和 $\gcd(15, 2^{\frac{4}{2}} - 1)$。這表示 $N = 15$ 的兩個重要因子分別為 5 和×3 。

5.1 5 Shor 尋序演算法複雜度評估

在圖 5 .11 中，具有 n *個*量子位元的精度暫存器（第一個暫存器或高位元暫存器）表示 X 模 N 的階數 r 。因為 r 的值小於或等於 N *的值，所以* n 的值是大於或等於 $\log_2 N$ *的最小整數*，我們可以將其寫成 $n = \lceil \log_2 N \rceil$。圖 5 .11 中， L *個量子*位元的工作暫存器（第二個暫存器或低位元暫存器）用於儲存 $X^P \pmod N$ 為 0≤ 計算的結果 $p \le 2n\ 1$. 因為 $-X^P \pmod N$ 的計算結果為 $0 \le p \le 2n - 1$ 小於 N *的值，* L 的值是大於或等於 $\log_2 N$ *的最小整數*，我們可以寫成 $L = \lceil \log_2 N \rceil$。 X 的值小於 N *的值*，因此我們可以用 L *位元來表示* X 的值和 N 的值。

在圖 5.11 中，在平行模型中，它同時計算每個模冪 $X^P \pmod N$ 到 0≤ *磷* ≤2n –1. 在圖 5.11 中，它應用了量子閘 $|X^P$ 模 $N> = |X^{p_1 \times 2^{n-1} + p_2 \times 2^{n-2} + \cdots + p_n \times 2^{n-n}}$ mod $N>$ 對精度暫存器和工作暫存器進行操作。計算模*冪* $XP \pmod N$ *)*的方法包含兩個階段。第一階段使用模乘法透過對 $X \pmod N$ 求平方來計算 $X^2 \pmod N$。然後，透過對 $X^2 \pmod N$ *) 求平方*來計算出 $X^4 \pmod N$。然後，它透過對 $X^4 \pmod N$ *) 進行平方來*計算 $X^8 \pmod N$。透過這種方式，它繼續計算出 $X^{2^k} \pmod N$ 對於所有 k 直到 $(n-1)$。

由於 $n = \lceil \log_2 N \rceil$ 且 $L = \lceil \log_2 N \rceil$，因此我們應用 $n = O(L)$。因此，在第一階段它總共完成了 $(n-1) = O(L)$ 平方運算。平方運算由乘法指令和模指令組成。乘法指令中的被乘數和乘數均為 X ，長度為 L。乘法指令中的乘積為 $(2 \times L)$ 位長，在模組化指令中被除數。模組化指令中的除數為 N ，長度為 $n = O(L)$ 位元。乘法指令中輔助進位的個數為（$2 \times L + 1$) 位，模指令中輔助借位位的個數為 $(2 \times L + 1)$ 位。

由於實現一條乘法指令的電路成本為 $O(L^2)$ 個數位邏輯門，實現一條模組化指令的電路成本為 O (*L 2) 個數位邏輯門，因此量子電路實現一條模組化指令的成本為* $O(L^2)$個數字邏輯閘。因此，完成的成本 $(n-1) = O(L)$ 平方運算是 $O(L^3)$ 個量子閘，實現第一階段的總成本是 $O(L^3)$ 個量子閘。

接下來，方法的第二階段是完成 $X^P \pmod N = (X^{p_1 \times 2^{n-1}}$ (模 N)) $\times_N$ ($X^{p_2 \times 2^{n-2}} \pmod N$) $\times_N \ldots \times_N$ ($X^{p_n \times 2^{n-n}} \pmod N$)。它實現了（ $n-1) = O(L)$ 次

模乘。每個模乘法由乘法指令和模指令組成。乘法指令中的被乘數和乘數都是 L 位長。乘法指令中的乘積為 $(2\times L)$ 位長，在模組化指令中被除數。模組化指令中的除數為 N ，長度為 $n=O(L)$ 位元。乘法指令中輔助進位的個數為（$2\times L+1$) 位，模指令中輔助借位位的個數為 $(2\times L+1)$ 位。

由於實現一條乘法指令的電路成本為 $O(L^2)$ 個數位邏輯門，實現一條模組化指令的電路成本為 $O(L^2)$ *個數位邏輯門，因此量子電路實現一條模組化指令的成本為* $O(L^2)$*個數位*邏輯閘。因此，完成的成本 $(n-1)=O(L)$ 模乘法是 $O(L^3)$ 個量子閘，實現第二階段的總成本是 $O(L^3)$ 個量子閘。這意味著實現量子電路的成本$|X^P$ 模 $N>=|X^{p_1\times 2^{n-1}+p_2\times 2^{n-2}+\cdots+p_n\times 2^{n-n}}$ mod $N>$ 對精度暫存器和工作暫存器進行操作的是 $O(L^3)$ 個量子閘。

接下來，在圖 5.11 中，它完成了一個逆量子傅立葉變換。實現一個量子傅立葉逆變換的成本是 $O(L^2)$ 個量子閘。最後，在圖 5.11 中，完成了對精度暫存器的測量。因此，在 Shor 的尋序演算法中，計算 X 模 N 的階 r *的成本*是 $O(L)$ 個量子位元和 $O(L^3)$ 個量子閘。

5.16 總結

在本章中，我們介紹了基礎數論。接下來，我們描述了**歐幾裡得**演算法。我們也引入了二次同餘。然後我們舉例說明了連分數。我們也介紹了找單和保理這兩個問題。接下來，我們描述如何計算 2 模 15 的階數以及 15 的質因數分解。如何計算 2 模 21 的階數和 21 的質因數分解。的*質因數*分解。我們也說明了公鑰密碼學和 RSA 密碼系統。接下來，我們介紹如何實現三個量子位元的***受控交換閘***。我們也描述了 Shor 的尋序演算法。然後我們說明如何設計 15 因式分解的量子電路。

5.17 參考文獻註釋

本章詳細介紹了數論的基礎知識和高級知識，推薦書籍 是 [哈代和賴特 1979；尼爾森和莊 2000； 伊姆雷和巴拉茲 200 5 ;利普頓和里根 20 14 ；席爾瓦 20 18；約翰斯頓等人，2019 年]。要更詳細地描述 Shor 的尋序演算法，推薦的文章和書籍是[Shor 1994;尼爾森和莊 2000； 伊姆雷和巴拉茲 200 5 ;利普頓和里根 20 14 ；席爾瓦 20 18；約翰斯頓等人，2019]。關於 Open QASM 指令的一個很好的介紹是 中的著名文章 [克羅斯等 2017]。

5.18 練習

5 .1 設 c 和 d 為整數，r *為* c 除以 d *的*餘數。然後提供了 $r \neq 0$，請證明方程式 gcd(c , d) = gcd(d , r)。

5.2（**中國剩餘定理**）我們假設 $y_1, \ldots, y_n$ 是正整數，使得任一對 y_i 和 y_j ($i \neq j$) 互質。那麼方程組

$$z = c_1 \text{ (模 } 1)$$
$$z = c_2 \text{ (模 } 2)$$
$$\ldots$$
$$z = c_n \text{ (mod } y_n)$$

有一個解決方案。此外，該方程組的任何兩個解都相等 Y 模 $= y_1 y_2 \ldots$ *是否*。請證明**中國剩餘定理**。

5.3 我們假設 p 和 k 是整數且 p *是1 到*p 範圍內的質數 –1. *然後*質數 p 除以 $\binom{p}{k}$。請證明一下。

5.4（**費馬小定理**）我們假設 p 是質數， a 是整數。則 $a^p = a \text{ (mod } p)$。如果整數 a 不能被質數 p *整除*，則 $a^{p-1} = 1 \text{ (mod } p)$。請證明他們。

5.5 *歐拉*函數 $\varphi(n)$ *定義為小於* n 且與 n *互質*的正整數的個數。我們假設 a *與* n 互質。則 $a^{\varphi(n)} = 1 \text{ (mod } n)$。

5.6 若 $(i / 2^n)$ 是有理分數且 z 和 r 是滿足 $|(z / r) - (i / 2^n)|$的正整數≤$(1 / (2 \times r^2))$，則 (z / r) *是* $(i / 2^n)$連分式的收斂。

5.7 證明$|1 - e^{\sqrt{-1} \times \theta}|^2 = 4 \times$罪$^2 (\theta / 2)$。