

國立台灣大學工學院工業工程學研究所

碩士論文

Institute of Industrial Engineering  
College of Engineering  
Nation Taiwan University  
Master Thesis

物理氣相沉積設備作業與順序控制邏輯之

彩色時間斐式網建模

Colored and Timed Petri Net-based operation sequence control logic  
modeling of physical vapor deposition (PVD) equipment

邱顯強

Shian-Chiang Chiou

指導教授：張時中 博士、范治民 博士

Advisor: Dr. Shi-Chung Chang, Dr. Chih-Min Fan

中華民國 九十九 年 七 月

July, 2010

~ i ~



## 誌謝

研究所兩年期間，很充實但也真的很辛苦，感謝指導教授張時中老師及范治民老師的諄諄教誨，付出許多的心力和時間來帶領我做研究，以及教導我正確的為學態度：研究之餘，亦不忘與我們分享人生經驗與待人處事的大道理，讓我永生難忘，使我不論在專業知識或生活處事上都受益良多，在此獻上最誠摯的謝意。並感謝口試委員：張展裕部經理、許丕敏課經理和范治民教授對於學生論文所提供的寶貴建議與指教，使得論文的內容更加完確。本論文部分係由台灣華亞科技股份有限公司計畫(Consulting Service on Dynamic Tool Allocation and Tool Sequencing Optimization: Modeling, Analysis and Simulation Methods, C10-0025)的支持下完成，特此致謝。

七百多個日子，時間飛快，幾乎不敢相信現在我要畢業了。這段日子感謝 109 的夥伴們冠迪、家瑜、小高及雅綾在有一餐沒一餐的聚餐下度過了生澀的碩一。碩二回到 207，家有四妻，酒量超好的輝哥、腦中無限幻想卻常崩潰的如因學姊、外冷內熱的籃球之神阿丞、有一就有二的猶豫達人佳慶、總是以減肥為口號的長腿女王小高，感謝妳對我的百般容忍、超酷重複條列式發言的酒醉小吉、感情生活充滿老梗愛裝可愛的緊身男憨憨、嚴重挑戰我貼心地位但滿腦外星思考的多食男墨西哥、正經八百好男人的庭育、總是默默在我右邊做事、睡覺、發呆的佳穎，這其中當然更不能缺少 Cindy、春瑤、方翔你們大家的加持，我們擁有共同的回憶與許多不能說的秘密，實驗室有大家真是多采多姿到亂七八糟。

最後，要感謝這些日子在我身邊不時提醒我放下手邊工作回家休息、擔心我畢不了業、永遠搞不清楚我在做什麼的爸媽，為了讓您相信我沒有在摸魚打混，這反倒是成為了激勵我把事情做完的動力，我由衷的感謝。

邱顯強 謹誌於

國立台灣大學工業工程研究所

中華民國九十九年七月

## 中文摘要

隨著半導體技術提升，現今 300mm 製程的半導體晶圓廠的投資額提升到了三十到四十億美元，其中約有 74% 是花在設備上。依據產業對整體設備效能(Overall Equipment Effectiveness, OEE)及特定機台效能(Intrinsic Equipment Effectiveness, IEE)的分析，發現未最佳化之瓶頸叢聚式設備會導致 OEE 喪失多達 40%，拖長產品生產時間而 IEE 偏低。進一步探討影響叢聚式設備效能的因素，設備的產出率(Throughput rate)及使用率(Utilization)往往深受作業的順序的影響。

為獲致有效的叢聚式設備作業順序，關鍵之一在於如何設計叢聚式設備中負責運送晶圓至反應室(Chamber)的手臂控制邏輯。回顧文獻發現斐式網被大量使用在於叢聚式設備建模，其中不乏針對晶圓生產流程進行斐式網建模(Hu, 1999)，也有針對特定手臂控制邏輯建模(Wu, 2006)，但缺乏整合兩者特色之斐式網建模方法。

本研究針對觀察到的叢聚式設備手臂控制邏輯提出一建模流程，此建模流程針對叢聚式設備之作業順序與手臂控制邏輯的彩色時間斐式網建模及其模組成，並以 PVD 設備作為本研究建模討論案例，最終目的是為了建構一個標準化的建模流程。在 PVD 設備作業上可分為 Loadlock、晶圓 process flow 及 Transfer 模組，手臂控制邏輯上可分為 Push 推式、Pull 拉式、FIFO 先進先出式控制邏輯模組，而主要所面對的挑戰為：(1) 如何將觀察到的叢聚式設備手臂控制邏輯建模，及(2) 如何將斐式網模組成。對此，本研究採用語意式的手臂控制邏輯建模概要建構模組化的彩色時間斐式網，並提出模組化的彩色時間斐式網合成演算法結合 PVD 設備作業模組、手臂控制邏輯模組，最後使用斐式網模擬環境 CPN tools® 進行模擬，針對合成完成之彩色時間斐式網比對實際作業機台進行效能分析，探討 (1) 晶圓產出時間間隔 (2) 手臂運輸觸發時間 及(3) 設備占用時間。

關鍵字：彩色時間斐式網、叢聚式設備、手臂控制邏輯建模、斐式網模組成

## **Abstract**

Capital investment of a 300mm semiconductor fab has grown to 3-4 billions USD as the technology evolves. Equipment cost per fab may count up to 74% of the capital investment. Individual tool performance has therefore been even more critical to fab productivity and competitiveness than before. Analysis of Overall Equipment Effectiveness (OEE) and Intrinsic Equipment Effectiveness (IEE) of tools by semiconductor fabs indicate that un-optimized constrained cluster tool performance may contribute up to 40% of OEE loss and production time waste in IEE loss. Among the factors that affect cluster tool performance the sequencing of intra-tool operations is the most important to the throughput rate and utilization.

For the semiconductor industry, Key to the operation sequence efficiency of a cluster tool is the control of robot operations, which transfer wafers from one chamber to another for processing. By reviewing of the literature, we noticed that Petri-Net is widely used in the cluster tool modeling, many of them consist of production process modeling using Petri Net (Hu, 1999), and some focus on specific arms control logic modeling (Wu, 2006), But lack of integration between both of the those characteristics of the Petri Net modeling.

In this paper, a standard modeling process for cluster tool sequence optimization under given robot control logics is established. The process consists of firstly using Colored and Timed Petri Net (CTPN) model cluster tool and associated operations modules, then conversion of CTPN to mathematical optimization problem. A Physical

Vapor Deposition (PVD) cluster tool is adopted as our study case. Our objective is to develop a standard modeling process for sequence optimization of cluster tools in general.

Modeling by using CTPN, the operations of a PVD cluster tool can be decomposed into three basic CTPN modules: Loadlock module, wafer process flow module and Transfer modules. Then, the Specific designing challenges are as following: C1) How to model control logics from the observed sequencing behaviors in the PVD tool such as Push, Pull, and first-in-first-out (FIFO) control logics? C2) How to synthesize all the modules into a complete sequencing model of the PVD tool? A two-step process is designed to address the two challenges.

In this research, we use semantic description methodology to interpret the arms control logic modules into CTPN model, and proposed a CTPN modular synthesis algorithm to generate and merge the PVD equipment operation module and arms control logic modules as a complete CTPN model, Finally using CPN tools @ simulation environment to simulate the PVD cluster tool model and validate the result by comparing with the actual machine performance, in this case we analyzed with following performance index: (1) wafer output time interval (2) the transporting arm's trigger time (3) the occupation time.

**Key word: CTPN, Petri Net, Cluster tool, Robot control logic, Synthesis**

# 目錄

口試委員會審定書 .....	i
誌謝 .....	ii
中文摘要 .....	iii
英文摘要 .....	iv
第一章 物理氣相沉積(PVD)設備作業與控制邏輯建模簡介 .....	1
1.1 PVD 設備作業與控制邏輯建模問題 .....	1
1.2 文獻探討 - 現有作業與控制邏輯建模方法 .....	2
1.3 研究範疇 .....	3
1.4 論文架構 .....	5
第二章 PVD 設備之斐式網建模 .....	6
2.1 一般 PVD 設備模組簡介 .....	7
2.1.1 運輸模組 (Transport Modules) .....	8
2.1.2 卡匣模組 (Cassette Loadlock Modules) .....	8
2.1.3 晶圓作業模組 ( Process Modules) .....	10
2.1.4 PVD 設備主體結構 .....	11
2.2 控制邏輯模組 - 手臂作業順序控制 .....	11
2.2.1 Push 推式手臂控制邏輯 .....	12
2.2.2 Pull 拉式手臂控制邏輯 .....	13
2.2.3 FIFO 先進先出式手臂控制邏輯 .....	14
2.3 PVD 設備建模工具 - 斐式網 .....	15
2.3.1 斐式網簡介 .....	15
2.3.2 現有斐式網建模 .....	15
2.4 研究挑戰 .....	16
第三章 手臂作業順序控制邏輯的時間斐式網建模 .....	17
3.1 彩色時間斐式網 .....	17
3.2 觀察之控制邏輯建模導引 .....	18
3.3 資源模組 .....	24
3.4 手臂控制邏輯模組 .....	27
3.5 晶圓生產流程 .....	28
第四章 彩色時間斐式網模組自動合成演算法及其模型驗證 .....	31
4.1 自動化斐式網模組合演算法 .....	31
4.2 模擬執行與初步驗證 .....	49
第五章 結論 .....	51
參考文獻 .....	54

## 圖目錄

圖 2.1 Endura 2 設備 (引用來源 Applied material [8]).....	7
圖 2.2 運輸模組(引用來源 Applied material [8]).....	8
圖 2.3 PVD 設備的卡匣模組.....	9
圖 2.4 PVD 製程所配置的反應室.....	10
圖 2.5 手臂發生 SWAP 時, 在甘特圖觀察到的特徵.....	12
圖 2.6 Push 推式手臂控制邏輯.....	13
圖 2.7 Pull 推式手臂控制邏輯.....	14
圖 3.1 語法式的手臂控制邏輯描述集合.....	20
圖 3.2 Push 控制器之語法式描述.....	20
圖 3.3 語法式的手臂控制邏輯描述所扮演的角色.....	20
圖 3.4 完整 Push 推式手臂控制邏輯.....	24
圖 3.5 幫浦之斐式網模組.....	25
圖 3.6 反應室產能斐式網.....	26
圖 3.7 手臂資源模組.....	26
圖 3.8 Push 推式手臂控制邏輯.....	27
圖 3.9 晶圓生產流程模組.....	30
圖 4.1 四步驟的模組成.....	40
圖 4.2 CPN tools <sup>®</sup> 模擬介面.....	49
圖 4.3 平均晶圓輸出時間間隔.....	50
圖 4.4 手臂隨著時間變化的行為預測(1).....	50
圖 4.5 手臂隨著時間變化的行為預測(2).....	50





# 第一章

## 物理氣相沉積(PVD)設備作業與控制邏輯建模簡介

### 1.1 PVD 設備作業與控制邏輯建模問題

半導體製造代工近年來致力於推動 12 吋晶圓製程(300mm)，在技術升級的驅使之下，單位建廠主要投資大幅成長 30 至 40 億美元，其中在設備投資上佔建廠總成本的比例更來到 74%的高成本投入[1]。正因此，設備的效能表現對於代工廠在產能上扮演著更為競爭的地位。

分析整體設備效能(Overall Equipment Effectiveness, OEE)及特定機台效能(Intrinsic Equipment Effectiveness, IEE)，半導體製造代工廠發現相較於尚未最佳化與執行最佳化方法之瓶頸叢聚式生產設備的效能表現上會導致 40% OEE 喪失 [2]，更顯著的反應在拖垮產品生產時間致使 IEE 偏低。深入探討影響叢聚式半導體生產設備效能的因素諸如：產出率(throughput rate)及設備使用率(Utilization)，叢聚式設備作業的順序扮演者很重要的要素。為了提供有效的叢聚式設備作業順序，最主要的關鍵在於如何有效率的控制叢聚式生產設備中的負責運送晶圓至各式反應室(chamber)的手臂控制邏輯。

有鑑於此，有效率之手臂控制邏輯將可以大幅提升叢聚式設備生產效能，有效提升 OEE 並減少單一叢聚式半導體生產設備生產時間。而叢聚式半導體生產設備往往具有資源共享所帶來的生產效率，但也往往導致資源競爭的問題，叢聚式生產設備中手臂資源的利用也是如此。然而，分析現有叢聚式半導體設備建模文獻中，研究學者多所著重於叢聚式半導體生產設備之生產流程建模，更應用斐式

網路建模方法模擬資源共享之競爭限制，卻忽略了叢聚式半導體生產設備在於生產作業時所採用之生產控制邏輯，往往受影響最深的是手臂的控制邏輯，因此，本研究將著手探討叢聚式半導體生產設備隻手臂控制建模方法並結合斐式網路可模擬多重資源共享產生之限制問題。

## 1.2 文獻探討 – 現有作業與控制邏輯建模方法

現有在叢聚式半導體生產設備之作業與控制邏輯建模方法，有非常多的建模方法與最佳化作業順序求解理論。深入討叢聚式設備作業順序與控制邏輯問題，建模、最佳化求解方法，學術界及工業界都有提出一些解決方案 [3, 4]，其中應用斐式網路為基礎進行設備作業順序建模的為大多數 [5]，更有開放式斐式網路建模與模擬軟體 CPN Tools [6] 提供斐式網路建模與模擬所需。然而，大部分的模型都假設所有的其路徑控制邏輯都可任意設計並建構在這些叢聚式半導體生產設備上，也就是以叢聚式半導體設備製造者的觀點來探討問題與其解決方案。

以晶圓廠經營者的觀點來看，大多數手臂運輸的移動控制邏輯是在採購叢聚式半導體生產設備時已預先建構在系統軟/硬體之中，而這些預設的手臂移動控制邏輯時常被視為是無法更改的黑盒子[7]，以最佳化手臂作業順序的觀點來看，這些手臂控制邏輯都是未知的限制式。要如何針對觀測到的機械手臂運作為有效的建模並且建構優化叢聚式半導體生產設備的機械手臂移動路徑控制邏輯是對於半導體製造商是重要但卻未被關注的議題。

許多業界學者嘗試藉由觀察半導體生產設備中手臂控制邏輯、或是根據設備所提供之不同手臂控制邏輯來加以建模，其中以 Applied Material 所生產之叢聚式半導體生產設備最被廣泛的研究。使用斐式網路建構半導體生產設之控制邏輯，Wu 和 Zhuo [8] 提出了目前在叢聚式半導體生產設備中手臂控制邏輯最被

廣泛使用的：SWAP Operation，並針對此控制邏輯以斐式網方式建模，最後以零工式生產模式(Job Shop)做為手臂作業順序最佳化之規劃，分析其建模優缺點，該文獻嘗試著眼於手臂控制邏輯建模並加以最佳化，但卻必須根據特定的手臂控制邏輯加以客製化建模，缺乏一標準的手臂控制邏輯建模方式，導致在面臨作業順序最佳化的同時，其數學模型的轉換也必須客製化設計，耗時費力。

斐式網擁有完整的建模、分析、數學形式呈現規則。一當斐式網建構完成後，可即刻進行模擬及其效能評估，以利於驗證模型之正確性，針對個別的斐式網作業關係，可對應到其專屬的數學呈現，進而作為線性整數規劃求解中之限制式，透過套裝軟體來達到最佳化斐式網流程順序。本研究中，將觀察到的叢聚式設備手臂控制邏輯提出一建模流程，此建模流程針對叢聚式設備之作業順序與手臂控制邏輯的彩色時間斐式網建模及其模組成，並以 PVD 設備作為本研究建模討論案例，最終目的是為了建構一個標準化的建模流程。

### 1.3 研究範疇

本研究探討如何設計手臂控制邏輯進行建模與效能分析以做為叢聚式半導體生產設備作業順序最佳化之用，在文獻探討中我們發現到許多學者針對叢聚式半導體生產設備多所著墨，在斐式網的建模過程中唯獨缺乏手臂控制邏輯之模型建構；對此本研究為設計具有標準化流程的叢聚式半導體設備作業順序與手臂控制邏輯之斐式建模以便有效率的轉換並建構數學最佳化模型，下列三點為本研究所將面臨的挑戰，並於各章節中逐一論述、實證：

- D1 如何從觀察叢聚式半導體生產設備作業順序之特色進行手臂控制邏輯建模。
- D2 如何設計手臂控制邏輯模組與一般叢聚式設備作業模組間的合成方法。

對此，本研究透過使用高等斐式網針對物理氣相沉積之叢聚式半導體生產設備 Applied Material Endura 2® 作為手臂控制邏輯觀察實體，並將實體依照叢聚式半導體生產設備之物理特色將其模組化，詳細說明如下 M1~M2(對應 D1~D2) 所示：

M1 以彩色時間斐式網(CTPN)為基礎的叢聚式半導體生產設備作業之手臂控制邏輯建模。

- Pull 推式手臂控制邏輯, Push 拉式手臂控制邏輯 及 FIFO 先進先出式手臂控制邏輯。
- 以彩色時間斐式網為基礎的手臂控制邏輯建模綱要。

M2 兩子斐式網合成演算法結合四步驟彩色時間斐式網搜尋配對、合成演算法：將 (1) 晶圓生產流程模組 (2) 未使用手臂控制邏輯之資源模組 (3) 使用手臂控制邏輯之資源模型模組 與 (4) 手臂控制邏輯模組成。

遵循上述手臂控制邏輯模組設計綱要並於將其與一般叢聚式半導體生產設備作業模組成為一完整斐式網進而轉換成整數規劃數學模型限制式，將叢聚式半導體生產設備之作業順序與手臂控制邏輯建模、模擬及最佳化提供一有效的參考標準作業流程，詳細內容如下列 R1~R2(對應 M1~M2)所示：

R1 系統化且快速的時間斐式網用以建構叢聚式半導體生產設備之作業順序與手臂控制邏輯方法。

R2 模組為基礎的時間斐式網手臂控制邏輯建模及模型驗證。

在實作方面，以 Applied Material Endura 2® for PVD 作為研究實體，分析設備之作業順序與手臂控制邏輯建模，文末更將合成之斐式網模型導入業界實

際線上資料，比對其晶圓產出間隔時間(inter-晶圓 time)、資源占用時間(occupied time)及手臂經由控制邏輯操作後與實際機台手臂運作行為之預測。

最後，本文為學術界及業界所能帶來的貢獻為(S1~S2)如下所示：

S1 叢聚式半導體生產設備之作業順序最佳化可大幅提升 OEE、IEE 及晶圓廠產能。

S2 標準化流程可提供有系統的快速叢聚式半導體生產設備作業順序之最佳化模型建置，並可應用最佳化套裝工具求解。

## 1.4 論文架構

在此提醒讀者本研究陳述架構如下。第二章說明一般叢聚式半導體生產設備所具有的一般模組與手臂控制邏輯模組，說明現今使用斐式網建模的方法與本研究中所使用之斐式網建模方式，第三章針對本研究重點：手臂控制邏輯之斐式網建模與將觀察到之手臂控制邏輯建模綱要，提供一致性的建模流程，第四章提出合成演算法組合一般叢聚式半導體生產設備模組及手臂控制邏輯模組，並根據合成結果帶入業界實際線上資料，比對其晶圓產出間隔時間(inter-wafer time)、資源占用時間(occupied time)及手臂經由控制邏輯操作後與實際機台手臂運作行為之預測，第五章為本研究下結論並提出未來工作項目。

## 第二章

### PVD 設備之斐式網建模

為研究叢聚式半導體生產設備的機械手臂移動路徑問題，物理氣相沉積(PVD) 叢聚式設備被我們選用為研究資料的來源。物理氣相沉積設備常被用於薄膜沉積的製程，也是廣泛被用於超大型積體電路(VLSI circuit)生產的製程技術。物理氣相沉積所形成的薄膜提供設備以及絕緣體間的傳導能力並且保障其應有的其原有的傳導環境。也因為 PVD 技術提供了精準的傳導控制，均勻的薄膜厚度，以及對於半導體材料更佳的結晶結構，PVD 叢聚式半導體生產設備受到半導體廠廣泛的應用與推崇。

物理氣相沉積的叢聚式半導體生產設備是一個使用單片晶圓處理，多反應室製程的完全自動化的系統。在每個反應室中同一個時間點不可處理多於一片的晶圓。機械手臂負責從反應室中拿取完成所在反應室的程序的晶圓至下一個程序之反應室。晶圓的進入與輸出都是經由卡匣。整合各種不同的模組並讓他們能再不同的生產路徑上進行模擬並且考量到物理氣相沉積叢集式半導體生產設備晶圓在不同程序或不同製程階段中機械手臂、緩衝空間以及反應室的資源使用與競爭，大幅度的增加了運作的複雜度以及模擬成本。物理氣相沉積叢聚式半導體生產設備的輸出率是上述這些模組間複雜的程序交互影響所產生的結果。如何有效率的操作叢聚式半導體生產設備並不是一件簡單的事情。

本章，我們會簡明的介紹在 2.1 節提出的物理氣相沉積叢聚式半導體生產設備的配置，在 2.2 節中從其他研究資料中針對物理氣相沉積設計的斐式網模型。

也將指出這些叢聚式半導體生產設備的作業流程中困難以及需要挑戰創新的地方。在 2.3 節，我們將會對我們所提出的物理氣相沉積叢聚式半導體生產設備的生產流程以及效能分析系統的基本架構做出描述。

## 2.1 一般 PVD 設備模組簡介

多數物理氣相沉積(PVD)半導體生產設備的反應室設計容許我們設定一致性的製程參數以做出精確的控制[11]。主要的物理氣相沉積叢聚式半導體生產設備組件有主體結構，晶圓生產模組，運輸模組以及卡匣模組。這些模組在叢集式半導體生產設備中都扮演著特定的角色，也有機械性的相互連結形成一個整合環境以完成製程中的所有工作。圖 2.1 呈現物理氣相沉積半導體生產設備的主要組件。

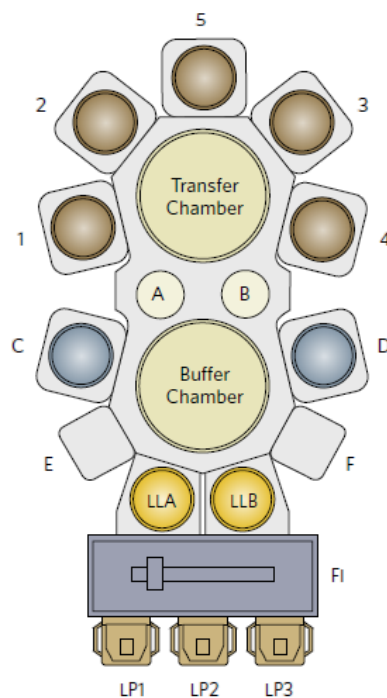


圖 2.1 Endura 2 設備 (引用來源 Applied material [8])



## 2.1.1 運輸模組 (Transport Modules)

在 PVD 機台中，運輸模組是由兩組 XP Robot(共四隻手臂，各自獨立運作)負責搬運機台中所有晶圓的機械手臂所扮演。一組手臂被配置在緩衝區間，另一組手臂在傳輸區間，每一組手臂都是由兩隻手臂所組成，分別負責上下兩層以此完全避免手臂知交叉碰撞等困擾，每隻手臂都有有四個行進方向，順時針旋轉、逆時針旋轉、手臂伸長以及縮短。圖 2.2 說明物理氣相沉積機械手臂的基本架構。而當手臂上並未抓取晶圓的時候，可以以較快的速度來進行移動。

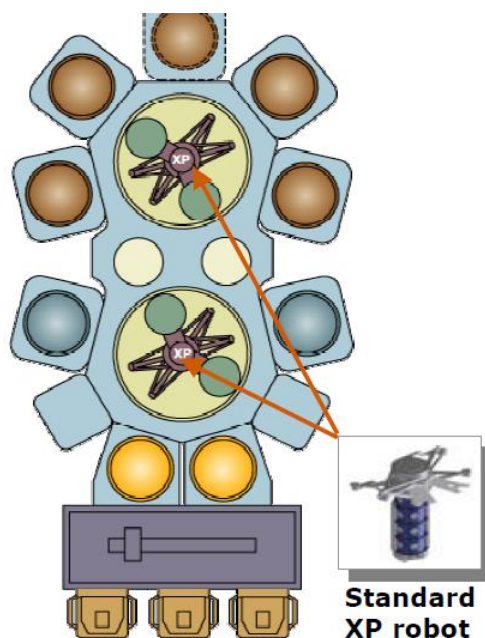


圖 2.2 運輸模組(引用來源 Applied material [8])

## 2.1.2 卡匣模組 (Cassette Loadlock Modules)

物理氣相沉積半導體生產設備通常有二至三個 Loadlock，這些卡匣提供系統一個讓晶圓與大氣隔絕的製程路徑一個管道，這允許主體結構由卡匣載入及卸載晶圓時能保持其在大氣壓力或大氣壓力以下控制環境的穩定，而晶圓要在幫浦

有能力處理的時候才能進入一切沉積的程序。而兩個 Loadlock 是分開且獨立的；也就是說，當一個 Loadlock 正在進行載入或卸載晶圓的時候並不影響另一個 Loadlock 的運作。圖 2.3 是一個在物理氣相沉積半導體生產設備的卡匣模組 [11]。卡匣模組是晶圓要進入主體結構的起始點，而晶圓必須經由物理氣相沉積半導體生產設備的卡匣模組進入製程。在標準的作業模式中，採用每個卡匣負載 25 片晶圓。

另外，為了提供一個晶圓進入和退出的管道，卡匣賦予了晶圓程序編排所需的儲存與檢索的能力，使兩個獨立作業的卡匣能編排各自的作業流程，包括了每片晶圓的獨特作業程序、平行生產以及反應室重複使用等。這些機制強化了流程彈性以及易用性並且進而改善了系統的效能。可規劃性、串行作業及平行作業的操作方式賦予了製程技術增加產量及彈性配置的可能性。

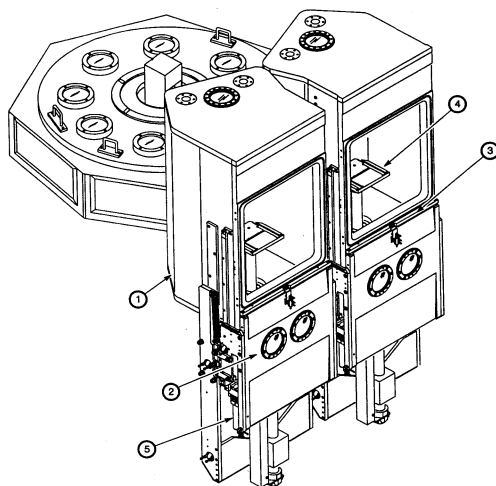


圖 2.3 PVD 設備的卡匣模組

### 2.1.3 晶圓作業模組 (Process Modules)

物理氣相沉積叢集式半導體生產設備採用每個反應室處理單一晶圓的方式來給予一個精確的控制處理環境。每一個反應室都有其特定的製程能力，依照不同的製程需求來分配設置系統中反應室數量與位置。圖 2.4 是一個典型的為物理氣相沉積製程所配置的反應室。

製程反應室可以任意附加在主體結構允許的模塊。而每個反應室都有其匣門(機蓋)以利於讓晶圓以機械手臂在反應室之間交互的移動。而每個匣門的開啟必須先經過壓力控制這個步驟。另外反應室也允許不同晶圓間程序上的改變，例如遇到不同的晶圓使用不同的製程時間，而這種轉換會需要額外的一些改變設置的時間。換句話說，一個反應室可以被許多不同製程的晶圓使用，但是不斷的轉換製程會使的設置時間變得很龐大而且大部分時候在轉換之後需要花時間與晶圓做測試，因此反應室通常都被指派處理特定的固定製程。

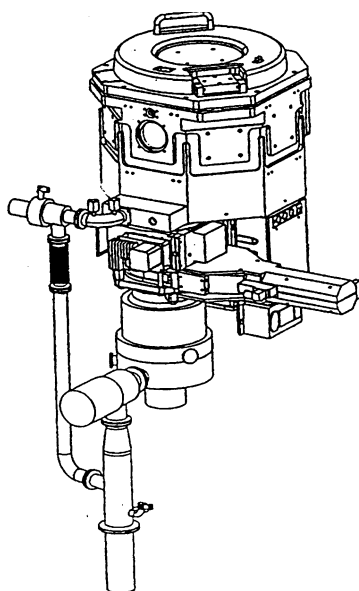


圖 2.4 PVD 製程所配置的反應室

## 2.1.4 PVD 設備主體結構

物理氣相沉積叢集式半導體生產工具的主體結構是一個鋁製的生產作業平台。然而它包含了兩個主要的反應室區間：緩衝區間和運輸區間。主體結構也是負載機械手臂的工作平台。兩組機械手臂位於兩個主要區間的中心以圓周的方式傳遞以及轉移晶圓。另一方面也是所有物理氣相沉積使用的反應室的載具，以模塊的方式讓所有的反應室能自由的依其需求做系統的整合。在運輸區間，主體結構提供了一到五個物理氣相沉積製程反應室的模塊，而在緩衝區間也提供了卡匣以及如化學氣相沉積、蝕刻等非物理氣相沉積所使用的製程模塊[12]

自由的製程反應室模組整合提供了許許多多的優勢，例如減少生產週期時間、減少傳輸時間，減低成本，減少汙染提高良率等等優良的製程控制及製程開發，大幅度的增加使用的靈活性；隨之而來的，也大幅度的增加了叢聚式半導體生產設備的操作複雜度，例如製程模組的安排限制，製程與業務上的優先順序對晶圓所造成的影響力等等。有太多的限制式必須滿足以得到一個良好可行的生產流程。

## 2.2 控制邏輯模組 - 手臂作業順序控制

除一般 PVD 叢聚式半導體生產設備相關之模組外，本研究重點著重在於手臂控制邏輯模組建模，在本節中將介紹：(1) Push 推式手臂控制邏輯、(2) Pull 拉式手臂控制邏輯 及 (3) FIFO 先進先出式手臂控制邏輯。

## 2.2.1 Push 推式手臂控制邏輯

Push 推式手臂控制邏輯，又稱雙機械手臂控制邏輯，顧名思義在 Push 推式手臂控制邏輯運作時，同時間必須控制兩隻手臂進行作業。此手臂控制邏輯的主要設計目的是為了減少叢聚式半導體設備之反應室閘門開啟次數並減少機械手臂呼叫時間，進而提高整體生產效率。而 Push 推式主要代表的是指此手臂控制邏輯的觸發時間點；假設一叢聚式半導體生產設備之晶圓作業次序為反應室 A → 反應室 B → 反應室 C，初始狀態為反應室 A、B 正在作業且反應室 C 為閒置狀態，此時 Push 推式的概念即為當反應室 A 預前往下一站反應室 B 時，反應室 B 的晶圓作業完畢後才會被機械手臂取出，反之晶圓則留在反應室 B 中繼續等待反應室 A 的往前推動。

為釐清手臂運作行為，業界一般使用甘特圖來觀看其手臂運作特色，Push 推式手臂控制邏輯中最大的特色就是會發生手臂 SWAP 行為，在甘特圖之時間軸當中我們可以發現如圖 2.5 所示：

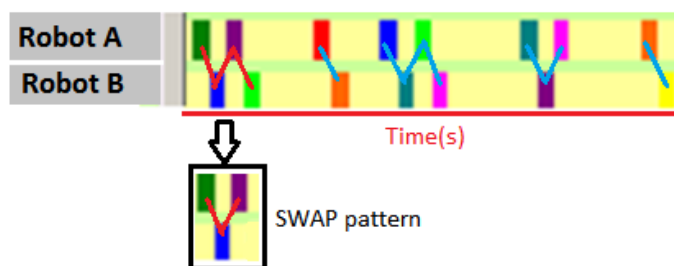


圖 2.5 手臂發生 SWAP 時，在甘特圖觀察到的特徵

而手臂運作的邏輯如下圖 2.6 所示，當反應室 A 完成作業時觸發 Push 推式手臂控制邏輯，在同時間控制機械手臂 X 與機械手臂 Y 依序執行下列步驟：(1) 機械手臂 X 從反應室 A 中取出晶圓至反應室 B 閘門口等待其作業完成 (2) 機械手臂 Y 直接前往反應室 B 中等待其作業完成 (3) 反應室 B 閘門開啟，機械手臂 Y

將晶圓從反應室 B 中取出後機械手臂 X 隨即將晶圓放入反應室 B 中，我們又稱第三步驟為 SWAP 作業。同樣步驟會一連貫的發生到所有反應室皆交換完成，即宣告此手臂控制邏輯的中止。

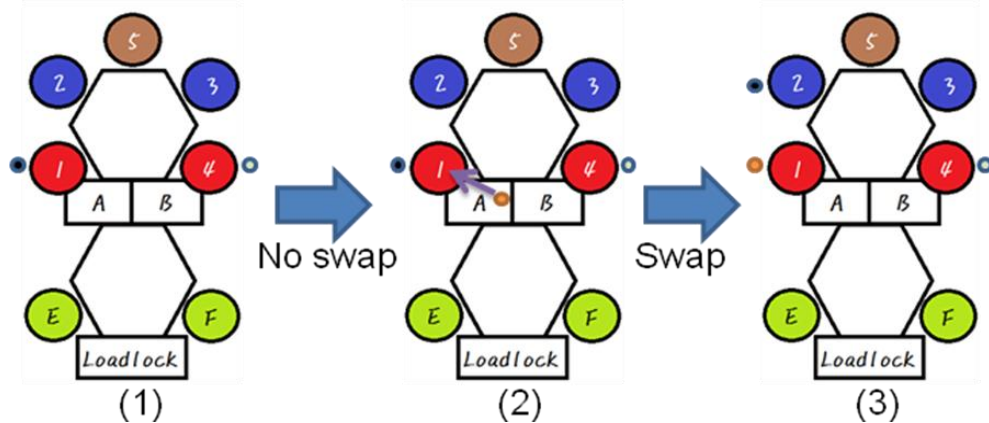


圖 2.6 Push 推式手臂控制邏輯

## 2.2.2 Pull 拉式手臂控制邏輯

Pull 拉式手臂控制邏輯，為單一手臂控制邏輯，此手臂控制邏輯的主要設計目的是為了減少晶圓在反應室中之等待時間，每當反應室一完成作業程序時，不需要檢查上一站反應室的作業完成與否，只需確認是否有單一機械手臂閒置。一經確認有單一機械手臂閒置，反應室會立即呼叫閒置手臂前往取走晶圓；因此時常會發生晶圓停留在手臂上等待下階段反應室作業，實務上第一片晶圓作業完成之時間往往比 Push 推式手臂控制邏輯來的快，但是觀察整體一批晶圓(Lot)的輸出時間，Push 推式手臂控制邏輯的作業完成時間往往會比 Pull 手臂控制邏輯來的短。

圖 2.7 中，我們利用簡單的手臂運作流程簡單說明 Pull 手臂控制邏輯的詳細運作情形：

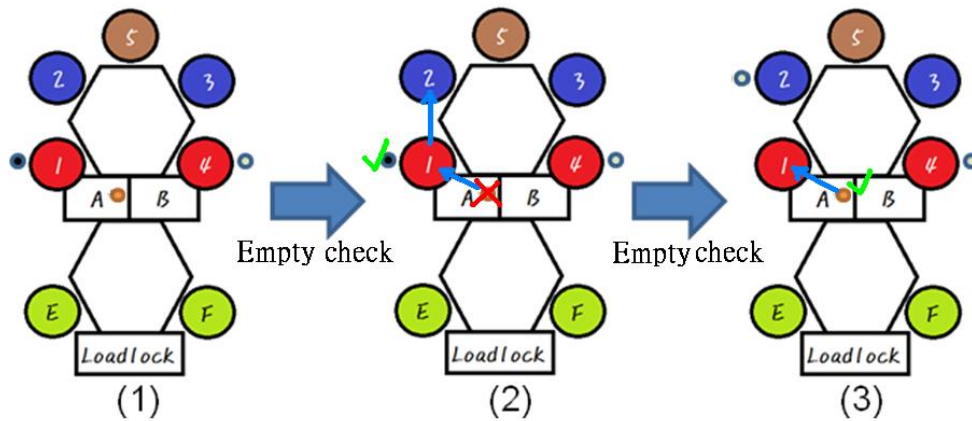


圖 2.7 Pull 推式手臂控制邏輯

### 2.2.3 FIFO 先進先出式手臂控制邏輯

先進先出式手臂控制邏輯為 PVD 設備中必須具備的控制邏輯，同時間存在於 Push 推式手臂控制邏輯與 Pull 拉式手臂控制邏輯當中，此處所提及之先進先出的概念係指在當 PVD 設備運作時，會有不同階段的製程需求。先進先出式手臂控制邏輯大致上可以分別從兩個觀點上面解釋：

(1) 相同作業反應室之先進先出式手臂控制邏輯：

Applied Endura® 2 的晶圓生產模組設定當中，可能會發生一到多個反應室所從事的加工作業是相同的。因此首先執行此項加工作業的晶圓必先離開此加工步驟。

(2) 整體 PVD 設備晶圓作業之先進先出式手臂控制邏輯：

顧名思義，整體 PVD 設備在運作的同時，晶圓作業完畢欲往下一站反應室作業時，或是晶圓已經完成作業與送入 Loadlock 進行 Venting 的時候，其作業優先順序均大於要進入該反應室或是晶圓正準備進入 Loadlock 進行 Pumping 的作業。

## 2.3 PVD 設備建模工具 – 斐式網

本文中，使用斐氏網來建構 PVD 設備的模型，這也是在 PVD 設備研究中最常被採用的模型方法之一。因為斐式網能以簡單直觀的圖示表達複雜的數學結構與限制，而市面上也有許多授權軟體提供模擬的需求，促使我們可以破除了數學抽象的限制，經由斐式網的建構與轉換，達到能以實際的物理觀察建立複雜的 PVD 設備數學模型，再進而利用套裝軟體尋找優化的可能。這種方式不但確保數學模型的正確性也可以確認是否有真實的呈現出機台的運作模式。

### 2.3.1 斐式網簡介

斐式網是對離散平行系統的數學表示。斐式網是 1960 年代由 Carl Adam Petri 發明的，既有嚴格的數學表述方式，也有直觀的圖形表達方式。由於斐式網能表達併發的事件，被認為是自動化理論的一種。而斐式網能夠很清楚的描述 Place 和 Transition 之間的因果關係，並由此構造時序。經典的斐式網主要式由四個元素所構成，Place, Transition, Arc 和 Marking, Place 描述 Marking 的狀態和位置，可以容納任意數量的 Marking, Transition 描述 Marking 的轉換變化，當一個 Transition 得到了所有 input Place 的 Marking，我們將這個情況稱為 Enable，而當 Transition 發生後，Marking 被消耗轉換到 output Place 的現象我們稱之為 Fire，而 Arc 則是 Place 和 Marking 兩者之間的因果關係，在基本的定義上，Place 和 Place 之間或 Transition 與 Transition 之間是不能有 Arc 的，而 Arc 也必須定義其方向性，Marking 則是所描述的動態對象，可以從一個 Place 經由 Transition 的觸發轉換到另一個 Place。

### 2.3.2 現有斐式網建模

斐氏網在平行運算，生產控制，網路通訊，交通運輸等都有廣泛的應用，因



為可以描述許多流程、因果、衝突、缺乏判斷與一致性等問題。也因為被廣泛的應用，最基本的斐式網無法漸漸無法滿足工業界的各種需求，進而加入了顏色以及時間元素。時間元素讓斐氏網能輕易的對整個模型的效能做合理的評估，顏色則是讓 Marking 有所區隔以做不同的處理。

## 2.4 研究挑戰

在前人的研究中，斐式網的建構已經能表達大部份的現象了；但是，在我們所使用的建模案例中，有兩個層面並沒有被直接的解決或深入的探討。第一個是特殊控制邏輯的建構，在機台中我們發現了特別的手臂運作控制模式，然而若是沒有合理的建構處理，只依靠晶圓流程模型來描述對於模擬得到的結果會造成極大的誤差。第二個是在時間斐式網中 Enable 和 Fire 的嚴格定義尤其是對於大量 Transition 同時被 Enable 的部分。然而，這兩個特點也將在本文之中被視為重要的挑戰並且詳加探討與說明。

## 第三章

### 手臂作業順序控制邏輯的時間斐式網建模

#### 3.1 彩色時間斐式網

斐氏網常以一個離散二維的表格來定義，該表格涵蓋了 Place, Transition 還有直接連結他們的 Arc。以圖形來說，Place 以圓圈表示，Transition 以長條或者方塊表示，而直接連結 Place 和 Transition 的 Arc 則是由箭頭表示，其中 Arc 又分為輸入 Arc 以及輸出 Arc，對應到其關係是進入 Place 還是離開 Place。而 Place 可以容納由黑點表示的 Marking；當然，Place 只能容納正數的 Marking 或者零個 Marking。這些 Marking, places 和 transitions 必須賦予他們在模型中所代表的獨特意義，在這篇論文中之定義如下(第四章有詳細說明)。

- Place 代表整個叢聚式半導體生產設備中晶圓的狀態、手臂的狀態及設備狀態。比方說，製程模組中的閒置時間或手臂移動的狀態，這些在模型中都被化作特定的 Place。在任何時間點上，Marking 描繪出生產系統當時的狀態：如在晶圓生產流程模組中的 Marking 可解讀為特定一片的晶圓狀態、在資源模組中的 Marking 可解讀成手臂或是反應室狀態。
- Transition 代表著一個系統製造行為，如反應室開始作業。而這個活動開始與結束在兩個連續的事件，Transition 的觸發意味者所有的前置 Place 皆已具備所需求之 Marking 始能觸發。
- Arc 直接連結的 Place 與 Transitions 的 Arc 代表著狀態也代表著在運作系統中的繼承的關係，如果有個 Arc 直接從 Place 到達一個 Transition，

我們將他視作入口 Place 對於這個 transition，入口 Transition 對於 Transition 是否可以釋出晶片扮演著重要的角色，只有當所有的入口 Place 都被滿足的時候，Transition 才有可能被觸發。

- Color 表示在同一個時間所看到的不同種類的資源，而在斐式網上固定的時間區間上我們可以定義  $k$  為相同的時間間隔，也說明了什麼是工作的生產時間。

### 3.2 觀察之控制邏輯建模導引

本研究中，我們針對可利用電腦優化工具套件求解的叢聚式半導體生產設備作業流程模型提出了一個標準快速並系統化的建構方法。為了達到這個目的，要如何對一個未知的控制邏輯或者設備的行為模式建構斐式網模型一直是個重大的挑戰，一般而言建模者會使用 WWNAWWA(有需要就加入)方法直接對機械手臂控制邏輯的觀察在時間斐式網中建構控制邏輯模型。為了達到快速建構優化模型這個目的，必須要有一個精確的規則或導引來幫助模型建構者或者工程師能以最快的方式構築其所需之控制邏輯。

在本章裡，我們首先模糊化機械手臂控制邏輯來已達到能以語法式描述其邏輯的目的，接著，將此這些語法式的描述分解成許多片段的控制單元，如資源分配、資源暫存、動作啟動與關閉等等的控制單元；最後再將這些片段的控制單元描述以 CTPN 的方式表現在模型中。

因此，此建構方法對於完全陌生的工程人員有深入淺出的引導作用：從最起初的物理作業邏輯的了解上就必須清楚的釐清設備間的關係，進而從學習到的物理邏輯運作透過簡而易懂的語法示描述呈現，搭配既存之控制邏輯參考下，組合出符合現況之控制邏輯，對於新手工程師熟悉此套方法僅須了解：

- (1) 機台實際物理作業意義，及
- (2) 何謂斐式網(3.1 詳述)

將可達到使用此套手臂控制邏輯斐式網建模之進入門檻，所花費時間即為工程師學習該項機台設備之花費時間。

### 3.2.1 語法式手臂控制邏輯描述

小強是一個工程師，有一天，台灣大學半導體製造公司的生產部門正在執行一個針對叢集式半導體生產設備利用 CTPN 的優化計畫，經過觀察，工程師小強發現他們現有的模型中並未考量到機械手臂的運作模式，而這個因素導致了模擬出來的結果與現實的資訊誤差。因此，小強的老闆請約翰以最快的速度將叢集式半導體生產設備的機械手臂控制邏輯模型建構出來。但是，小強最初對如何完成這個工作完全沒有概念。如果小強知道語法式的手臂控制邏輯描述這個方法，將可以幫助他釐清他所觀察到的手臂控制邏輯以及如何將他觀察到的事件正確的建構到一個時間裴氏網模型中。

為了產生手臂控制邏輯的時間裴氏網模型，工程師首先要描述他所觀察到的手臂行為模式到如圖 3.1 所示的語法式的手臂控制邏輯描述集合。語法式的手臂控制邏輯描述著重於文法式的規則，例如**主詞+動詞+受詞**的結構描述，這個方法可以幫助工程師以及建模者釐清他所觀察到的現象，作為串聯物理特性及 CTPN 模型的橋樑(如圖 3.3 所示)，以將他的觀察手動建置到時間裴式網模型中。在圖 3.2 提出的 Push 控制器可以做為一個以語法式描述來建構觀察到的現象的例子。

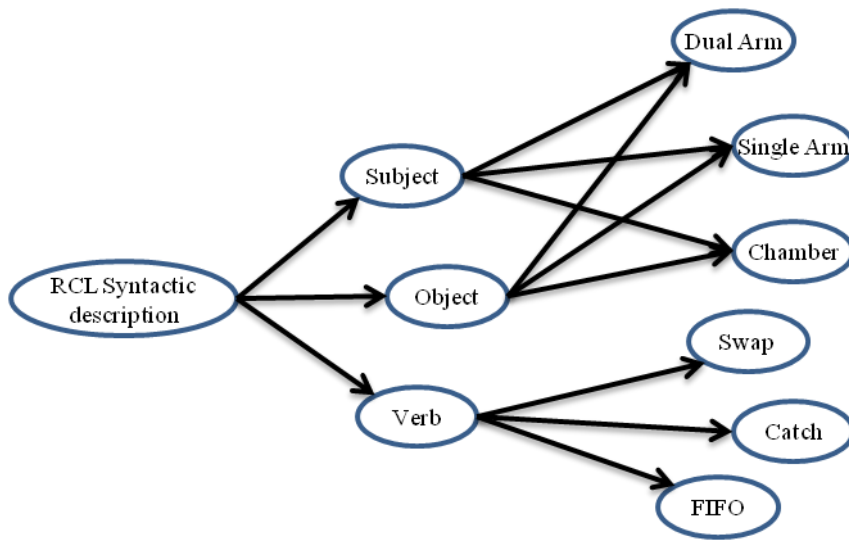


圖 3.1 語法式的手臂控制邏輯描述集合

Push Control logic: **Dual Arm** in **specific chamber k** with **SWAP** rule

圖 3.2 Push 控制器之語法式描述

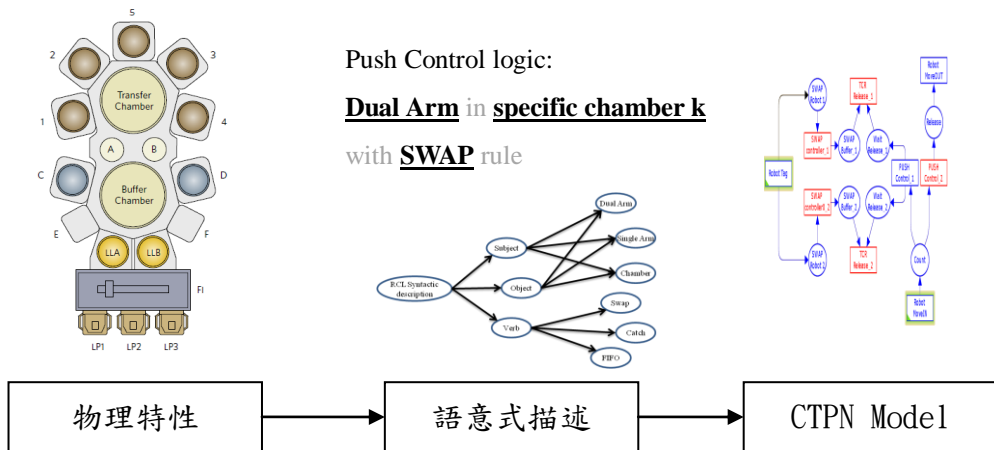


圖 3.3 語法式的手臂控制邏輯描述所扮演的角色

### 3.2.2 手臂控制邏輯功能性拆解

根據工程師的語法式描述，我們可以將這些描述分成特定的型別如驅動器、啟動條件、關閉條件和動作事件以此幫助工程師更深入的描述手臂控制邏輯，以下我們簡單的介紹其中元素：

#### 驅動器

在語法式描述中的主詞，主要描述在手臂控制邏輯運行中將要被控制的資源。例如在 Push 的控制邏輯中，兩個手臂在這被視做驅動器，也就是說這兩個手臂將會在控制邏輯運行的過程中被控制。

#### 啟動條件

在語法式描述中扮演著動詞的角色，主要描述在什麼樣的情況下驅動器應該要使用哪個特定的行為模式，我們使用一階邏輯(First order logic)來幫我們描述這些條件，例如(主詞狀態)  $\cap$  (主詞狀態)  $\cap \dots \rightarrow$  啟動。比方說 Push 手臂控制邏輯：(位置  $k$  製程結束)  $\cap$  (位置  $k+1$  不在等待狀態)  $\rightarrow$  手臂控制邏輯(啟動)。

#### 關閉條件

在語法式描述中扮演著受詞的角色，主要描述在什麼樣的情況下驅動器應該要停止使用哪個特定的行為模式，我們同樣使用一階邏輯來幫我們描述這些條件，例如(主詞狀態)  $\cap$  (主詞狀態)  $\cap \dots \rightarrow$  關閉。比方說 Push 手臂控制邏輯：(位置  $k$  製程結束)  $\cap$  (位置  $k+1$  並不在被使用中)  $\rightarrow$  手臂控制邏輯(啟動)。

## 動作事件

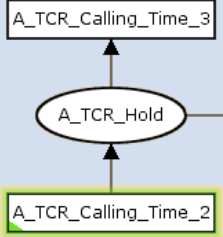
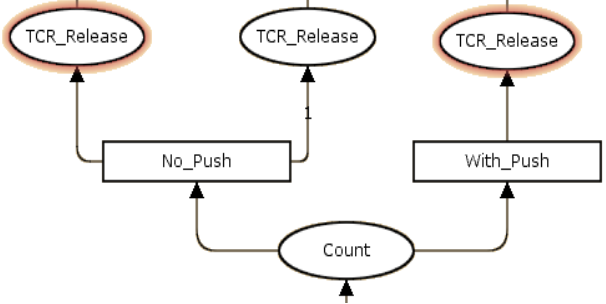
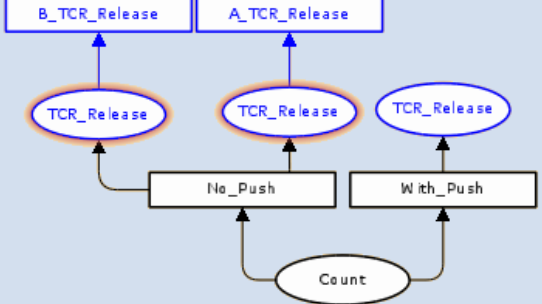
主詞，受詞和動詞的語法式描述結構，工程師一部一部的詳細描述手臂控制邏輯的動作事件，應該為當啟動條件滿足時應當做的連續且具體的行動事件。每個動作事件也都該以一階邏輯描述的方式建構，比方說(手臂控制邏輯 (箝制狀態))  $\cap$  (驅動器動作 (晶片, 目標物)  $\rightarrow$  做動作(事件).，以 Push 推式手臂控制邏輯為例：

(1) (手臂控制邏輯 (啟動))  $\cap$  (驅動器 (抓取 (晶片, 位置  $k$ )))  $\cap \rightarrow$  行動 (步驟一),

(2) (手臂控制邏輯 (行動(步驟一)))  $\cap$  (驅動器(抓取(晶片, 位置  $k+1$ )))  $\cap \rightarrow$  行動(步驟二),

(3) (手臂控制邏輯 (行動(步驟一)))  $\cap$  (驅動器(放置 (晶片, 位置  $k+1$ )))  $\cap \rightarrow$  行動(步驟三)

根據這種方式的分類，我們可以進一步依據驅動器，啟動條件，關閉條件以及行動事件等種類在時間裴式網中建構出資源選擇，計數器以及 transition 觸發功能等模型，以 Push 手臂控制邏輯為例在表格一可以看到：

Classification of syntactic description	Functions of CTPN	Example of Push RCL
<b>Actor</b>	Resource cuing function	
<b>Enabling / Disable Condition</b>	Token counter function	
<b>Do event</b>	Transition firing function	

最後，我們結合這些以功能為基礎的時間裴式網模型到一個完整的 Push 推式手臂控制邏輯的時間裴式網模型中如所示。



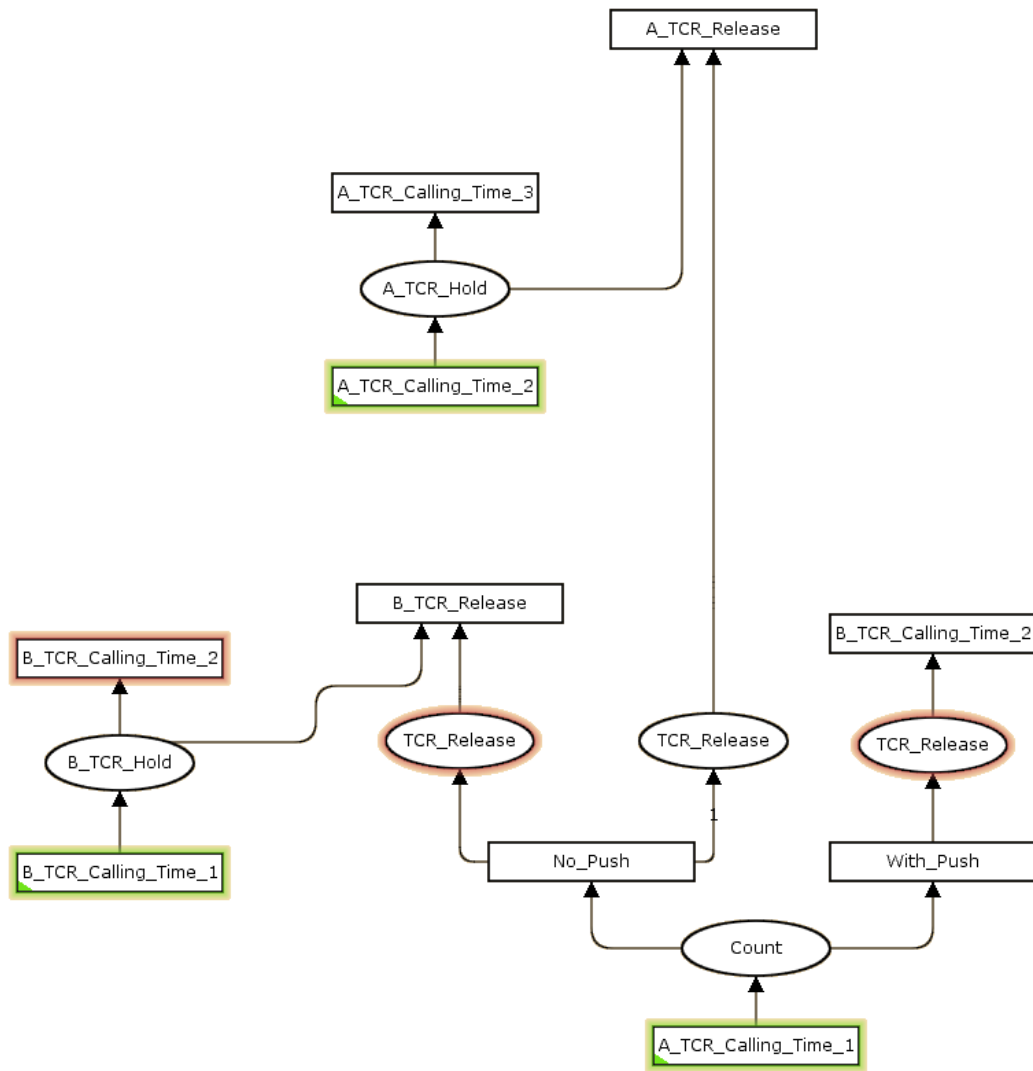


圖 3.4 完整 Push 推式手臂控制邏輯

### 3.3 資源模組

本研究針對資源模組，分為兩大類型：(1) 有使用手臂控制邏輯之資源模組 (如幫浦資源、反應室資源及反應室產能) (2) 未使用手臂控制邏輯之資源模組 (如緩衝區間手臂資源及傳輸區間手臂資源)；以下將分別介紹各模組之斐式網設計。

## 幫浦

主要應用在於 Loadlock 中，Applied Endura® 2 最多可具有三個獨立式的 Loadlock 但卻需共用同樣一個的幫浦作為 Pumping 時所用，唯一有限量之資源，圖 3.5 為幫浦之斐式網。

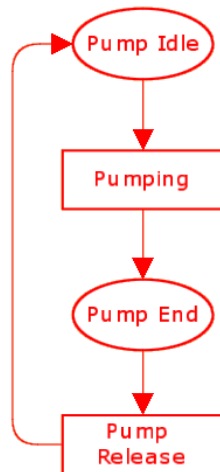


圖 3.5 幫浦之斐式網模組

## 反應室產能

反應室通常最多只能夠容納一片晶圓在裡面進行作業，為了避免在斐式網進行模擬時出現同一 Place 具有多個 Marking 發生，導致單一反應室卻出現兩片以上(含)的晶圓在裡面的情形發生，因此反應室產能可視為一有限資源，下圖 3.6 紅色部分即為反應室產能斐式網。

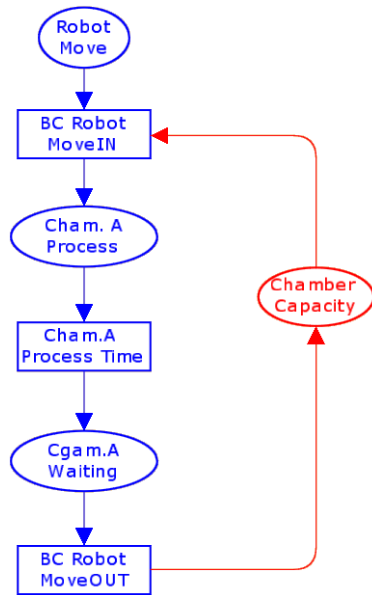


圖 3.6 反應室產能斐式網

### 手臂

主要應用在於晶圓出入反應室時，夾取晶圓之用，通常與 3.4 手臂控制邏輯具有互動，下圖 3.7 之手臂資源中會隨著手臂名稱的不同而隨之更改，在 Applied Endura® 2 中，我們總共會有四隻手臂供我們使用，分別是：Ruffer chamber robot 1 及 2 和 Transfer chamber 1 及 2。

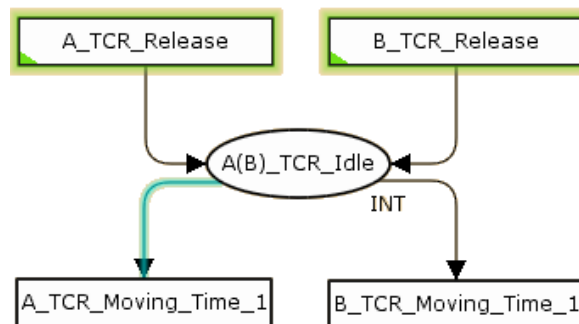


圖 3.7 手臂資源模組

### 3.4 手臂控制邏輯模組

手臂控制邏輯部分，根據由 PVD 設備所觀察到的手臂運作現象，並藉由 3.2 節中手臂控制邏輯的建模導引協助下，我們可以順利的將其轉換成特定的斐式網，下列將逐一介紹手臂控制邏輯斐式網。

#### Push 推式控制邏輯

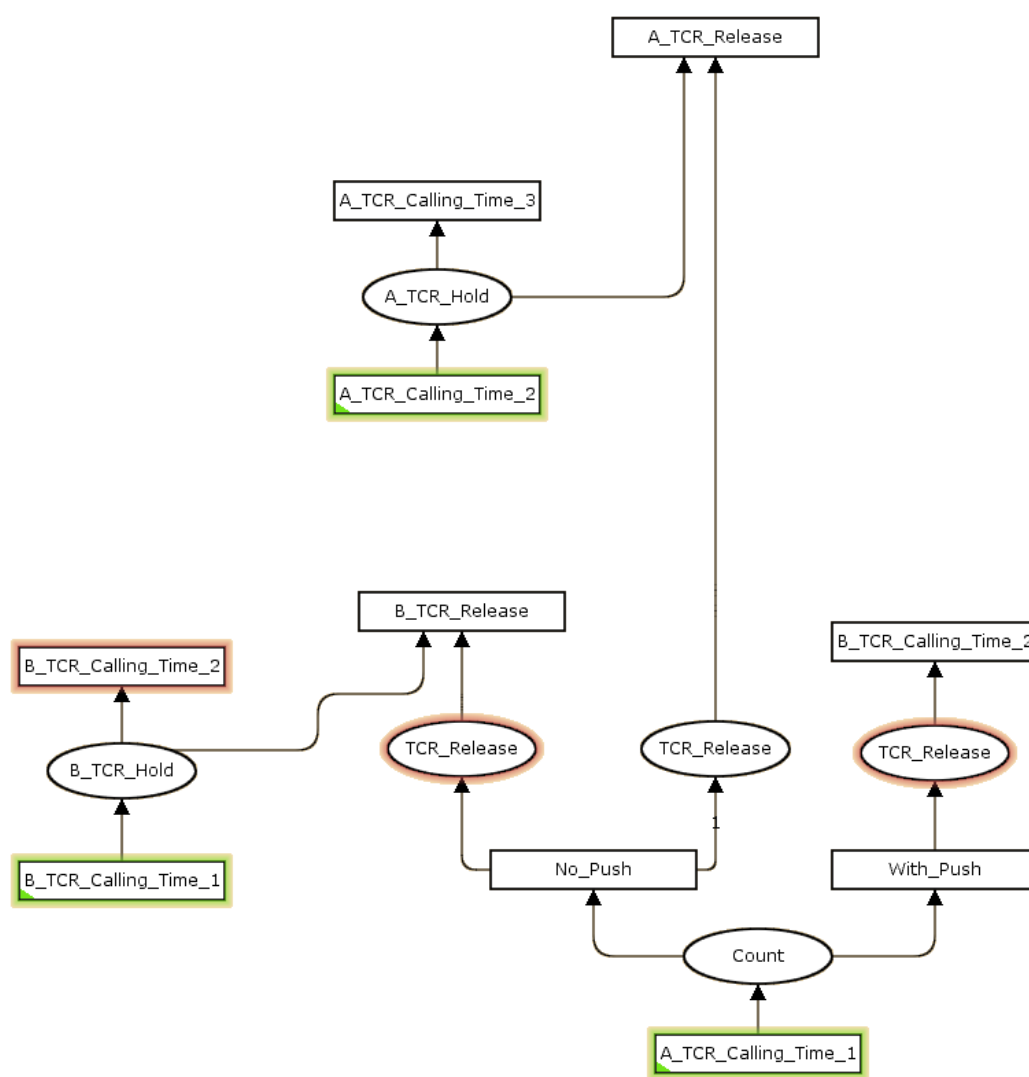


圖 3.8 Push 推式手臂控制邏輯

## Pull 拉式控制邏輯

即傳統手臂工作邏輯，無需使用任何額外手臂控制邏輯模組外加干涉，隨著斐式網模擬的特色，只要手臂資源允許，特定 Transition 被觸發後會持續的往下一反應室工作站前進，因此斐式網自然的可以忠實呈現此一現象。

## FIFO 先進先出式控制邏輯

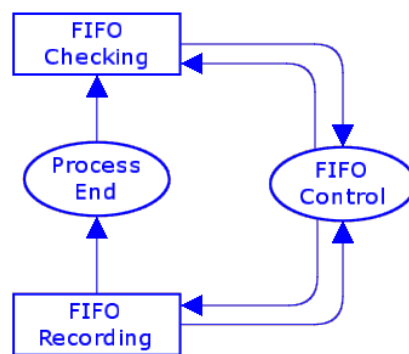


圖 3.9 FIFO 先進先出式控制邏輯

## 3.5 晶圓生產流程

晶圓生產流程式 PVD 設備斐式網建模的核心，資源模組、手臂控制邏輯模組均是依靠晶圓生產流程進行合成的動作，因此在將模组合成之前，我們必須優先定義晶圓生產流程，Microsoft Excel 是一個生產與製造部門最常用的工具之一，透過大家熟悉的介面維護晶圓生產流程資料是再適合不過的方法，表 3.1 中晶圓生產流程規範、手臂運作邏輯表，清楚的呈現晶圓生產的步驟、所耗費的時間、使用到的資源及手臂控制邏輯，該表之詳細定義如表 3.2 中所示。

表 3.1 晶圓生產流程規範、手臂運作邏輯表

RECIPE TYPE	STEP NO	STEP NAME	RECIPE TIME	CHAMBER NAME	CHAMBER STATE	ROBOT	ROBOT CONFIG	REROBOT SERVER MODE
RECIPEA	1	LOADIN	0	LOADPORT:A	ACTIVE	NONE	NONE	NONE
RECIPEA	2	PRE_ALNGR	8	LOADPORT:A	ACTIVE	FIROBOT	NONE	CONT
RECIPEA	3	PUMP	25	LOADLOCK:A:2	ACTIVE	FIROBOT	NONE	BOTH
RECIPEA	3	PUMP	25	LOADLOCK:B:2	ACTIVE	FIROBOT	NONE	BOTH
RECIPEA	4	DEGAS	28	CHAMBER:E	ACTIVE	FIROBOT	PUSH	INPUT
RECIPEA	4	DEGAS	28	CHAMBER:E	ACTIVE	BCROBOT:1	PUSH	OUTPUT
RECIPEA	4	DEGAS	28	CHAMBER:F	INACTIVE	FIROBOT	PUSH	INPUT
RECIPEA	4	DEGAS	28	CHAMBER:F	INACTIVE	BCROBOT:1	PUSH	OUTPUT
RECIPEA	5	PASS	0	CHAMBERA	ACTIVE	BCROBOT:1	PUSH	INPUT
RECIPEA	5	PASS	0	CHAMBERA	ACTIVE	TCROBOT:1	PUSH	OUTPUT

表 3.2 晶圓生產流程規範表欄位說明

欄位名稱	欄位意義
<b>RECIPE TYPE</b>	不同配方會有構成不同晶圓生產流程模組，本研究所提及之範例均採用單一配方
<b>STEP NO</b>	表示晶圓生產流程之順序關係， $[\text{STEP NO}] \in \mathbb{N}^+$
<b>STEP NAME</b>	晶圓生產流程中各步驟名稱
<b>RECIPE TIME</b>	晶圓生產流程中各步驟所耗費之時間理論值
<b>CHAMBER NAME</b>	晶圓生產流程所使用的資源模組
<b>CHAMBER STATE</b>	晶圓生產流程所使用的資源(反應室)模組使用狀態 $[\text{CHAMBER STATE}] \in \{\text{ACTIVE}, \text{INACTIVE}\}$
<b>ROBOT</b>	晶圓生產流程中，所使用到的資源(手臂)模組
<b>ROBOT CONFIG</b>	資源(手臂)模組所採用的手臂控制邏輯模組
<b>ROBOT SERVER MODE</b>	資源(手臂)模組對於晶圓生產流程的作用時間

透過表 3.1 的內容轉換，我們可以清楚的制定出晶圓生產流程  $Q=\{q_1 \rightarrow q_2 \rightarrow q_3 \cdots, q_{|Q} \}$ ，下圖 3.10 我們可以看到單一晶圓生產流程，並藉由比對晶圓生產流程  $Q$  後在該斐式網後向下串聯下去。



圖 3.10 晶圓生產流程模組

## 第四章

### 彩色時間斐式網模組自動合成演算法及其模型驗證

工業界生產步調緊湊，如何在有限的時間內創造出最大產能是大家所追求的目標。叢聚式半導體生產設備可根據不同晶圓製造流程做出調整，為此，本研究藉由建構模組化的斐式網模型，達到模擬模型可因應反應室變更、參數調整或配方(Recipe)改變的時候，在短時間內建構出其對應的斐式網模擬模型，進而執行設備效能評估。

第三章我們詳細介紹了叢聚式半導體生產設備的模組化設計及其建構概要。然而，單有這些模組，並不能保證模組間可快速組合成一個忠實呈現完整叢聚式半導體設備的斐氏網模型。設計自動化的模組成方法，將有助於消除傳統藉由人力將模組成所造成的錯誤疏失，並加速完整模擬模型的間模速度及模型的正確性。

本章節旨在根據過往學者所整理出合成規則與方法[10]，建構一自動化斐式網模組成演算法。分別在 4.1 節中說明兩子斐式網合成演算法，從演算法符號定義到搜尋應合成模組，探討演算法編寫流程及介紹相對應之程式偽代碼，並以一個小例子詳細說明整個斐式網合成演算法流程，4.2 節針對此演算法產生之工業界實際作業機台斐式網進行效能分析，驗證斐氏網是否正確及其應有之特性。

#### 4.1 自動化斐式網模組成演算法

一個完整的叢聚式半導體生產設備之斐式網模型，必須將其所相對應之模組進行合成，本節中首先在 4.1.1 提出如何將兩子斐式網模型進行合成，4.1.2 說



明完整斐式網合成時所經歷的步驟，最後在 4.1.3 中使用簡易的叢聚式半導體生產設備斐式網進行模組間的合成與完整斐式網的建構。

首先，本研究之演算法所將使用到的符號與變數定義如下列所示：

#### 符號定義

- $j$  : 未使用手臂控制邏輯之資源斐式網模型註標， $j \in \{ \_resource\_module \}$
- $s$  : 使用手臂控制邏輯之資源斐式網模型註標， $s \in \{ \_cl\_resource\_module \}$
- $x$  : 手臂控制邏輯斐式網模型註標， $x \in \{ \_cl\_module \}$
- $f$  : 晶圓生產流程斐式網模型註標， $f \in \{ \_process\_module \}$
- $P$  : 斐式網中 Places 所成的集合， $P = \{ p_u, u = 1, 2, \dots, |P| \}$ ， $|P|$  為該斐式網 Place 個數
- $T$  : 斐式網中 Transitions 所成的集合， $T = \{ t_v, v = 1, 2, \dots, |T| \}$ ， $|T|$  為該斐式網 Transition 個數
- $k$  : 時間註標，意指第  $k$  秒的開始， $k = 0, 1, 2, \dots, K$ ，其中  $k \in \mathbb{N}^+$
- $\bar{m}_k$  : 為  $|P| \times 1$  的行向量  $[m_{1k}, m_{2k}, \dots, m_{uk}, \dots, m_{|P|k}]^T$  其中  $m_{uk}$  表  $p_u$  在第  $k$  秒開始時之 Marking 數目
- $\bar{m}_0$  : 當  $k=0$  時稱為初始 Marking，為  $|P| \times 1$  的行向量  $[m_{10}, m_{20}, \dots, m_{u0}, \dots, m_{|P|0}]^T$
- $B^-$  : 為  $|P| \times |T|$  的矩陣， $B^- \equiv [b_{uv}^-], b_{uv}^- \equiv \begin{cases} 1, & \text{if } p_u \rightarrow t_v \text{ (輸入 } t_v \text{ 的 arc)} \\ 0, & \text{其他} \end{cases}$
- $B^+$  : 為  $|P| \times |T|$  的矩陣， $B^+ \equiv [b_{uv}^+], b_{uv}^+ \equiv \begin{cases} 1, & \text{if } t_v \rightarrow p_u \text{ (輸出 } t_v \text{ 的 arc)} \\ 0, & \text{其他} \end{cases}$
- $\Theta$  : 為  $|T| \times 1$  的行向量， $\Theta = [\theta_1, \theta_2, \dots, \theta_v, \dots, \theta_{|T|}]^T$ ，其中  $\theta_v$  為  $t_v$  觸發後的時間延遲

### 4.1.1 兩子斐式網合成演算法

在模組化斐式網中，我們將一完整之斐式網拆解成 (1) 晶圓生產流程模組 (2) 使用控制邏輯之資源模組 (3) 未使用控制邏輯之資源模組 及 (4) 手臂控制邏輯模組。不同模組雖具有不同物理意義，但其合成方法卻是相同。本節中介紹如何合成兩子斐式網。

合成，Hsieh 和 Chang 整理歸納指出在斐式網中，合成在圖像上的主要概念就是將擁有相同 Transition 命名之子斐式網結合[10]，文中指出一斐式網合成運算元為" $\parallel$ "，其定義如下所示：

$$\text{New-net} = \text{Sub-net A} \parallel \text{Sub-net B}$$

$\parallel$ : 合成運算元，搜尋合成運算元右方的子斐式網  $\text{Sub-net A} = \{ P_A, T_A, B_A^-, B_A^+, \Theta_A, M_{A0} \}$ ，與左方的子斐式網  $\text{Sub-net B} = \{ P_B, T_B, B_B^-, B_B^+, \Theta_B, M_{B0} \}$  之共同 Transition 將其合併為新的斐式網  $\text{New-net N} = \{ P_N, T_N, B_N^-, B_N^+, \Theta_N, M_{N0} \}$ 。合成的步驟可大略分成以下三大步驟：

Step 1 搜尋兩子斐式網之共同 Transition  $T_A \cap T_B \neq \{\Phi\}$

Step 2 合成兩子斐式網

1. 將 Transition 及 place 取聯集  $T_N = T_A \cup T_B, P_N = P_A \cup P_B$
2. 更新 Transition 交集部分之  $B_N^-, B_N^+$
3. 根據現有  $N = \{ P_N, T_N, B_N^-, B_N^+ \}$ ，更新  $\{ \Theta_N, M_{N0} \}$

### Step 3 完成合併

下列為詳述之兩子斐式網合成演算法：

輸入

1. 子斐式網路  $\text{SubNetA} = \{ P_A, T_A, B_A^-, B_A^+, \Theta_A, M_{A0} \}$
2. 子斐式網路  $\text{SubNetB} = \{ P_B, T_B, B_B^-, B_B^+, \Theta_B, M_{B0} \}$

輸出

1. 合成完之斐式網路  $\text{Net} = \{ P_N, T_N, B_N^-, B_N^+, \Theta_N, M_{N0} \}$

#### 二子斐式網路合成演算法

Step1 搜尋兩子斐式網之共同 Transition

If  $T_A \cap T_B \neq \{\Phi\}$  then

Go to step2

Else

Go to step4

End if

Step2 合成兩子斐式網

Update six-tuples of Petri Net

$$\begin{aligned} T_N &= T_A \cup T_B && // T_A = \{t_{Av}, v'=1,2,\dots, |T_A|\} \\ &= \{T_A \cap \overline{T_B}, T_A \cap T_B, \overline{T_A} \cap T_B\} && // T_B = \{t_{Bv'}, v''=1,2,\dots, |T_B|\} \\ &= \{t_{Nv}, v=1,2,\dots, |T_A \cap \overline{T_B}|, |T_A \cap \overline{T_B}|+1, \dots, |T_A \cap T_B|, |T_A \cap T_B|+1, \\ &\quad \dots, |T_A \cap \overline{T_B}|\} \end{aligned}$$

$$\begin{aligned} P_N &= P_A \cup P_B && // P_A = \{p_{Au'}, u'=1,2,\dots, |P_A|\} \\ &= \{P_A \cap \overline{P_B}, P_A \cap P_B, \overline{P_A} \cap P_B\} && // P_B = \{p_{Bu'}, u''=1,2,\dots, |P_B|\} \end{aligned}$$

$$= \{P_{Nu}, u=1,2,\dots, |P_A \cap \overline{P_B}|, |P_A \cap \overline{P_B}|+1, \dots, |P_A \cap P_B|, |P_A \cap P_B|+1, \dots, |P_A \cap \overline{P_B}|\}$$

```

New  $B_N^- = [0]_{|P_N| \times |T_N|}$  //初始化  $B_N^-$ 

New  $B_N^+ = [0]_{|P_N| \times |T_N|}$  //初始化  $B_N^+$ 

For each  $p_{Nv} \in P_N$  { //針對  $P_N$  中每個元素更新 Marking

    If  $p_{Nu} = p_{Au'}$  and  $p_{Nu} = p_{Bu'}$ , then //  $P_{Nu}$  同時存在  $P_A$  與  $P_B$  中 (交集)

         $m_{Nu0} = m_{Au'0} + m_{Bu'0}$  //Place 中的 Marking 相加

         $b_{Nu}^- = b_{Au'}^-$  //input arc 列向量

         $b_{Nu}^+ = b_{Au'}^+$  //output arc 列向量

    Else if  $p_{Nu} = p_{Au'}$ , then //  $P_{Nu}$  存在  $P_A$  (非交集)

         $m_{Nu0} = m_{Au'0}$  //取  $M_{A0}$  中 Place 的初始 Marking

         $b_{Nu}^- = b_{Au'}^-$  //將  $b_{Au'}^-$  填入初始化之  $B_N^-$ 

         $b_{Nu}^+ = b_{Au'}^+$  //將  $b_{Au'}^+$  填入初始化之  $B_N^+$ 

    Else //  $P_{Nu}$  存在  $P_B$  (非交集)

         $m_{Nu0} = m_{Bu'0}$  //取  $M_{B0}$  中 Place 的初始 Marking

         $b_{Nu}^- = b_{Bu'}^-$  //將  $b_{Bu'}^-$  填入初始化之  $B_N^-$ 

         $b_{Nu}^+ = b_{Bu'}^+$  //將  $b_{Bu'}^+$  填入初始化之  $B_N^+$ 

}

For each  $t_{Nv} \in T_N$  { // 針對  $T_N$  中每個元素更新時間延遲

    If  $t_{Nv} = t_{Av'}$  and  $t_{Nv} = t_{Bv'}$ , then { //  $t_{Nv}$  同時存在  $T_A$  與  $T_B$  中 (交集)

         $\theta_{Nv} = \theta_{Bv'}$  // 取 Transition 時間延遲的最大值

    Else if  $t_{Nv} = t_{Av'}$  //  $t_{Nv}$  僅存在  $T_A$  中 (非交集)

         $\theta_{Nv} = \theta_{Av'}$  // 取  $T_A$  中 Transition 時間延遲

```

```

Else // tNv 僅存在 TB 中 (非交集)
      θNv = θBv, // 取 TB 中 Transition 時間延遲
};
};

```

Step3 合成完畢

Net = { P<sub>N</sub>, T<sub>N</sub>, B<sub>N</sub><sup>-</sup>, B<sub>N</sub><sup>+</sup>, Θ<sub>N</sub>, M<sub>N0</sub>} 即合併之所得

Step4 終止合併

### 二子斐式網路合成演算法 - 合成範例

輸入

1. SubNetA = { P<sub>A</sub>, T<sub>A</sub>, B<sub>A</sub><sup>-</sup>, B<sub>A</sub><sup>+</sup>, Θ<sub>A</sub>, M<sub>A0</sub>}

$P_A = \{p_{A1}, p_{A2}, p_{A3}\} = \{ \text{TCR\_Moving}, \text{W\_Process}, \text{TCR\_Calling} \}$

$T_A = \{t_{A1}, t_{A2}\} = \{ \text{TCR\_Calling\_Time}, \text{W\_Process\_Time} \}$

$$B_A^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, B_A^+ = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, M_{A0} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\Theta_A = [0 \ 0]^T$$



$$2. \text{SubNetB} = \{ P_B, T_B, B_B^-, B_B^+, \Theta_B, M_{B0} \}$$

$$P_B = \{ p_{B1} \} = \{ W\_Chamber\_Idle \}$$

$$T_B = \{ t_{B1}, t_{B2} \} = \{ W\_Process\_Time, TCR\_Calling\_Time, \}$$

$$B_B^- = [0 \ 1], B_B^+ = [1 \ 0], M_{B0} = [1]$$

$$\Theta_B = [38 \ 5]^T$$



輸出

$$1. \text{合成完之斐式網路 Net} = \{ P_N, T_N, B_N^-, B_N^+, \Theta_N, M_{N0} \}$$

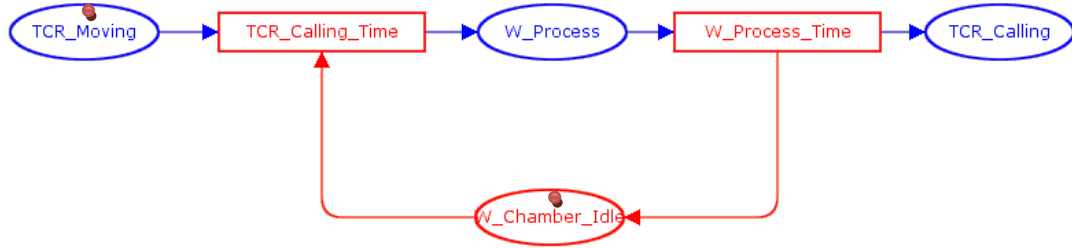
$$P_N = \{ p_{A1}, p_{A2}, p_{A3}, p_{B1} \} =$$

$$\{ TCR\_Moving, W\_Process, TCR\_Calling, W\_Chamber\_Idle \}$$

$$T_N = \{ t_{A1} = t_{B2}, t_{A2} = t_{B1} \} = \{ TCR\_Calling\_Time, W\_Process\_Time \}$$

$$M_{N0} = [1 \ 0 \ 0 \ 1]^T, \Theta_N = [5 \ 38]^T$$

$$\mathbf{B}_N^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \mathbf{B}_N^+ = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$



Step1 搜尋兩子斐式網之共同 Transition

$$T_A \cap T_B = \{t_{A1} = t_{B1}, t_{A2} = t_{B2}\} \neq \{\Phi\} \text{ then}$$

Go to Step2

Step2 合成兩子斐式網

$$T_N = \{T_A \cup T_B\} = \{t_{A1}, t_{A2}\} = \{TCR\_Calling\_Time, W\_Process\_Time\}$$

$$P_N = \{P_A \cup P_B\} = \{p_{A1}, p_{A2}, p_{A3}, p_{B1}\}$$

$$\text{New } \mathbf{B}_N^- = [ ]_{4 \times 2}, \mathbf{B}_N^- = \mathbf{0}$$

$$\text{New } \mathbf{B}_N^+ = [ ]_{4 \times 2}, \mathbf{B}_N^+ = \mathbf{0}$$

For each  $\{p_{A1}, p_{A2}, p_{A3}, p_{B1}\}$  {

$$p_{N1} = p_{A1}$$

$$\rightarrow m_{N10} = m_{A20} = 5$$

$$\rightarrow \mathbf{b}_{N1}^- = \mathbf{b}_{A1}^- = [0 \ 1]$$

$$\rightarrow \mathbf{b}_{N1}^+ = \mathbf{b}_{A1}^+ = [0 \ 0]$$

$$p_{N2} = p_{A2}$$

$$\rightarrow m_{N10} = m_{A10} = 0$$

$$\rightarrow \mathbf{b}_{N2}^- = \mathbf{b}_{A2}^- = [0 \ 1]$$

$$\rightarrow \mathbf{b}_{N2}^+ = \mathbf{b}_{A2}^+ = [1 \ 0]$$

$$P_{N3} = P_{A3}$$

$$\rightarrow m_{N30} = m_{A30} = 0$$

$$\rightarrow \mathbf{b}_{N3}^- = \mathbf{b}_{A3}^- = [0 \ 0]$$

$$\rightarrow \mathbf{b}_{N3}^+ = \mathbf{b}_{A3}^+ = [0 \ 1]$$

$$P_{N4} = P_{B1}$$

$$\rightarrow m_{N40} = m_{B10} = 38$$

$$\rightarrow \mathbf{b}_{N4}^- = \mathbf{b}_{B4}^- = [1 \ 0]$$

$$\rightarrow \mathbf{b}_{N4}^+ = \mathbf{b}_{A4}^+ = [0 \ 1]$$

$$\mathbf{B}_N^- = \begin{bmatrix} -\mathbf{b}_{N1}^- & - \\ -\mathbf{b}_{N2}^- & - \\ -\mathbf{b}_{N3}^- & - \\ -\mathbf{b}_{N4}^- & - \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \mathbf{B}_N^+ = \begin{bmatrix} -\mathbf{b}_{N1}^+ & - \\ -\mathbf{b}_{N2}^+ & - \\ -\mathbf{b}_{N3}^+ & - \\ -\mathbf{b}_{N4}^+ & - \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

};

Step3 合成完畢

Net = { P<sub>N</sub>, T<sub>N</sub>, B<sub>N</sub><sup>-</sup>, B<sub>N</sub><sup>+</sup>, Θ<sub>N</sub>, M<sub>N0</sub>} 即合併之所得

Step4 終止合併



### 4.1.2 斐式網模組配對搜尋演算法

一個完整忠實呈現叢聚式半導體生產設備的斐式網模型，必須將其所對應的模組進行合成。在此，我們將模組合成分為四步驟如下圖 4.1 所示：

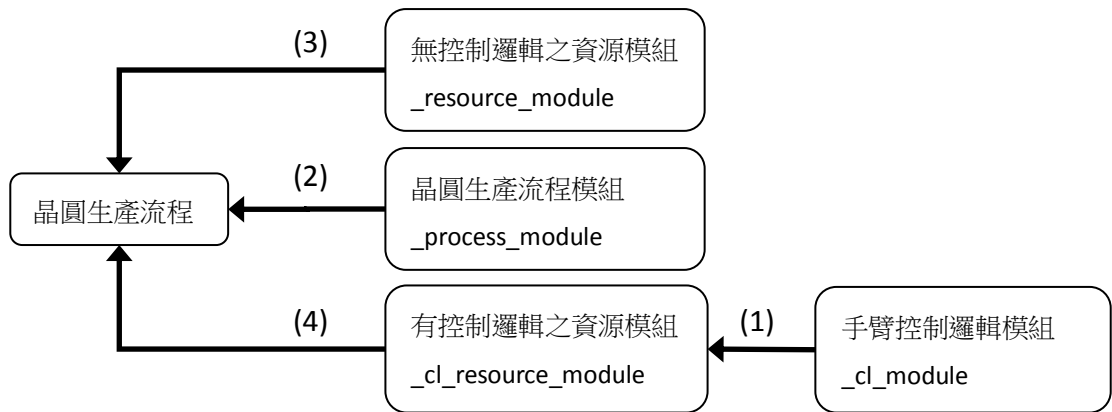


圖 4.1 四步驟的模組合成

首先定義最終合成完畢之斐式網為  $N = \{ P_N, T_N, B_N^-, B_N^+, \Theta_N, M_{N0} \}$ ,

(1)  $N_i =$  使用控制邏輯之資源模組 || 手臂控制邏輯模組

(2)  $N =$  晶圓生產流程模組 || 晶圓生產流程模組

(3)  $N =$  未使用控制邏輯之資源模組 ||  $N$

(4)  $N = N_i$  ||  $N$

上述四步驟合成的主要概念為：優先將使用手臂控制邏輯之資源模組與其相對應的手臂控制邏輯模組合成構成完整的資源模組，再分別將未使用控制邏輯之資源模組和已完成控制邏輯合成之使用手臂控制邏輯之資源模組與晶圓生產流程合成，進而產生完整的斐式網模型。本斐式網模組配對搜尋演算法之核心概念

即是根據四步驟合成的主要概念，搜尋符合模組類型之所有既存之斐式網模組是否具有共同 Transition，若有則合併，演算法中關於兩子斐式網合成運算元 || 詳見 4.1.1 節中之兩子斐式網合成演算法。

### 斐式網模組配對搜尋演算法

輸入

1. 未使用手臂控制邏輯之資源模組

$$RM_j = \{P_{RM_s}, T_{RM_s}, B_{RM_s}^-, B_{RM_s}^+, \Theta_{RM_s}, M_{RM_s,0}\}, j=1 \text{ to } |J|$$

2. 使用手臂控制邏輯之資源模組

$$CRM_s = \{P_{CRM_s}, T_{CRM_s}, B_{CRM_s}^-, B_{CRM_s}^+, \Theta_{CRM_s}, M_{CRM_s,0}\}, s=1 \text{ to } |S|$$

3. 手臂控制邏輯模組

$$CM_x = \{P_{CM_x}, T_{CM_x}, B_{CM_x}^-, B_{CM_x}^+, \Theta_{CM_x}, M_{CM_x,0}\}$$

4. 晶圓生產流程模組

$$WP_f = \{P_{WP_f}, T_{WP_f}, B_{WP_f}^-, B_{WP_f}^+, \Theta_{WP_f}, M_{WP_f,0}\}$$

輸出

1. 合成完畢之斐式網  $N = \{P_N, T_N, B_N^-, B_N^+, \Theta_N, M_{N,0}\}$

Step0. 輸入個別斐式網模組

1.  $RM_j = \{P_{RM_s}, T_{RM_s}, B_{RM_s}^-, B_{RM_s}^+, \Theta_{RM_s}, M_{RM_s,0}\}$
2.  $CRM_s = \{P_{CRM_s}, T_{CRM_s}, B_{CRM_s}^-, B_{CRM_s}^+, \Theta_{CRM_s}, M_{CRM_s,0}\}$
3.  $CM_x = \{P_{CM_x}, T_{CM_x}, B_{CM_x}^-, B_{CM_x}^+, \Theta_{CM_x}, M_{CM_x,0}\}$
4.  $WP_f = \{P_{WP_f}, T_{WP_f}, B_{WP_f}^-, B_{WP_f}^+, \Theta_{WP_f}, M_{WP_f,0}\}$

Step1. 合成 有使用手臂控制邏輯之資源模組 與 手臂控制邏輯模組

```
For each transition of CRMs { //TCRMs = {tCRMsv', v'=1,2,...,|TCRMs|}
    For each transition of CMx { //TCMx = {tCMxv, v=1,2,...,|TCMx|}
        If TCRMs ∩ TCMx ≠ Φ then
            N1 = CRMs || CMx // 表示有 CRMs有使用到 CMx
        End if
    };
};
```

Step2. 合成品圓生產流程模組

```
For each transition of WPf{ //TWPf = {tWPfv', v'=1,2,...,|TWPf|}
    N = N || WPf
};
```

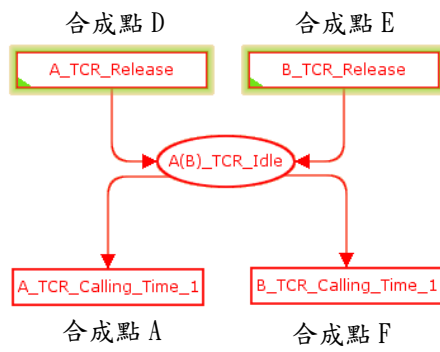
Step3. 合成 未使用控制邏輯之資源模組 與 Step2 產出之晶圓生產流程 N

```
For each transition of N { //TN = {tNv', v'=1,2,...,|TN|}
    For each transition of RMj { //TRMj = {tRMjv, v=1,2,...,|TRMj|}
        If TN ∩ TRMj ≠ Φ then
            N = N || RMj // 表示有 N有使用到 RMj
        End if
    };
};
```

Step4. 合成 Step1 產出之已含控制邏輯之資源模組 N1 與 Step3 產出之 N

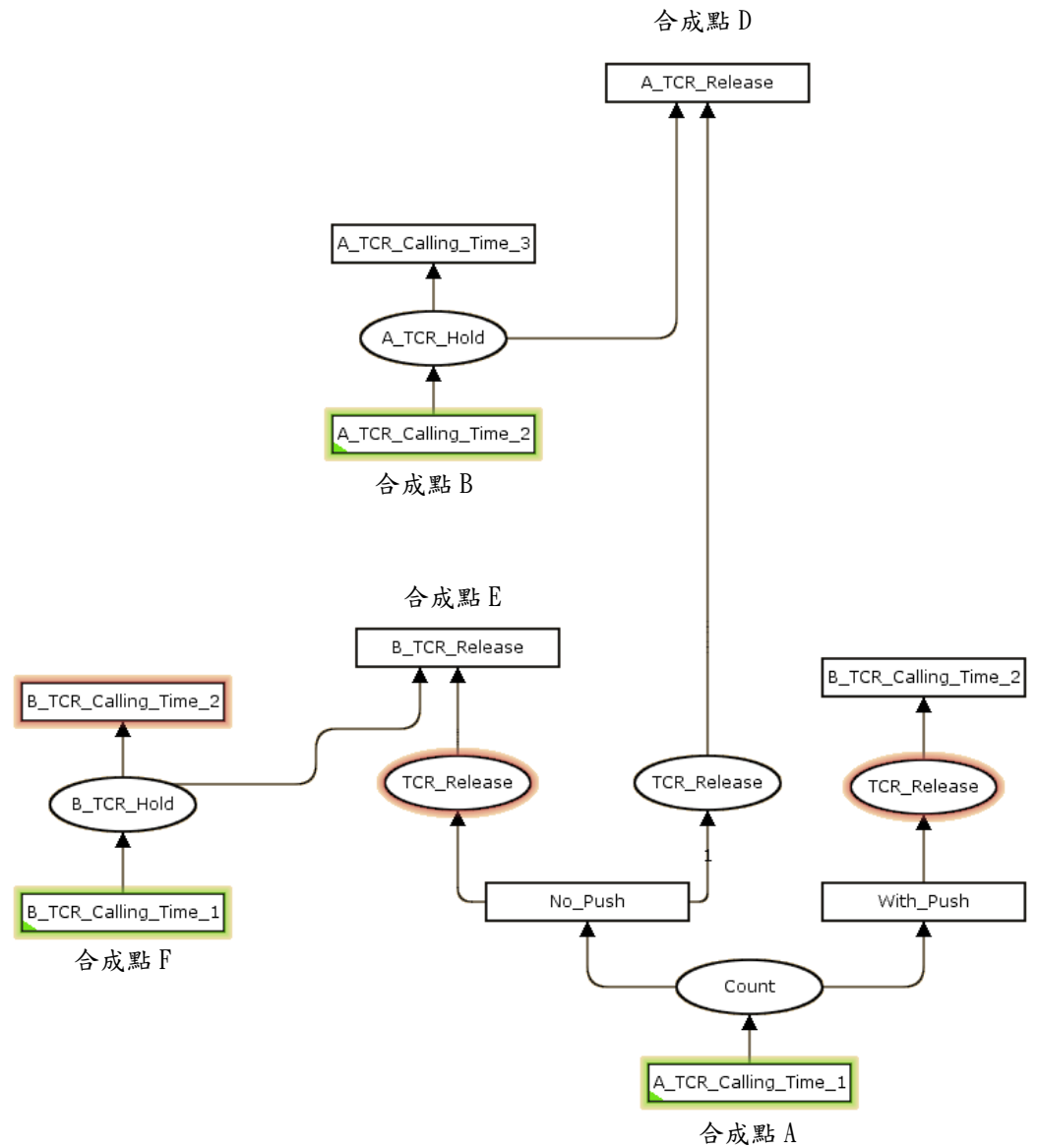
```
For each transition of N { //TN = {tNv', v'=1,2,...,|TN|}
```





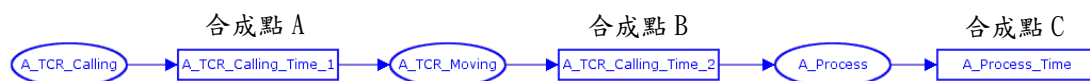
3. 手臂控制邏輯模組

$$CM_x = \{P_{CM_x}, T_{CM_x}, B_{CM_x}^-, B_{CM_x}^+, \Theta_{CM_x}, M_{CM_x,0}\}$$



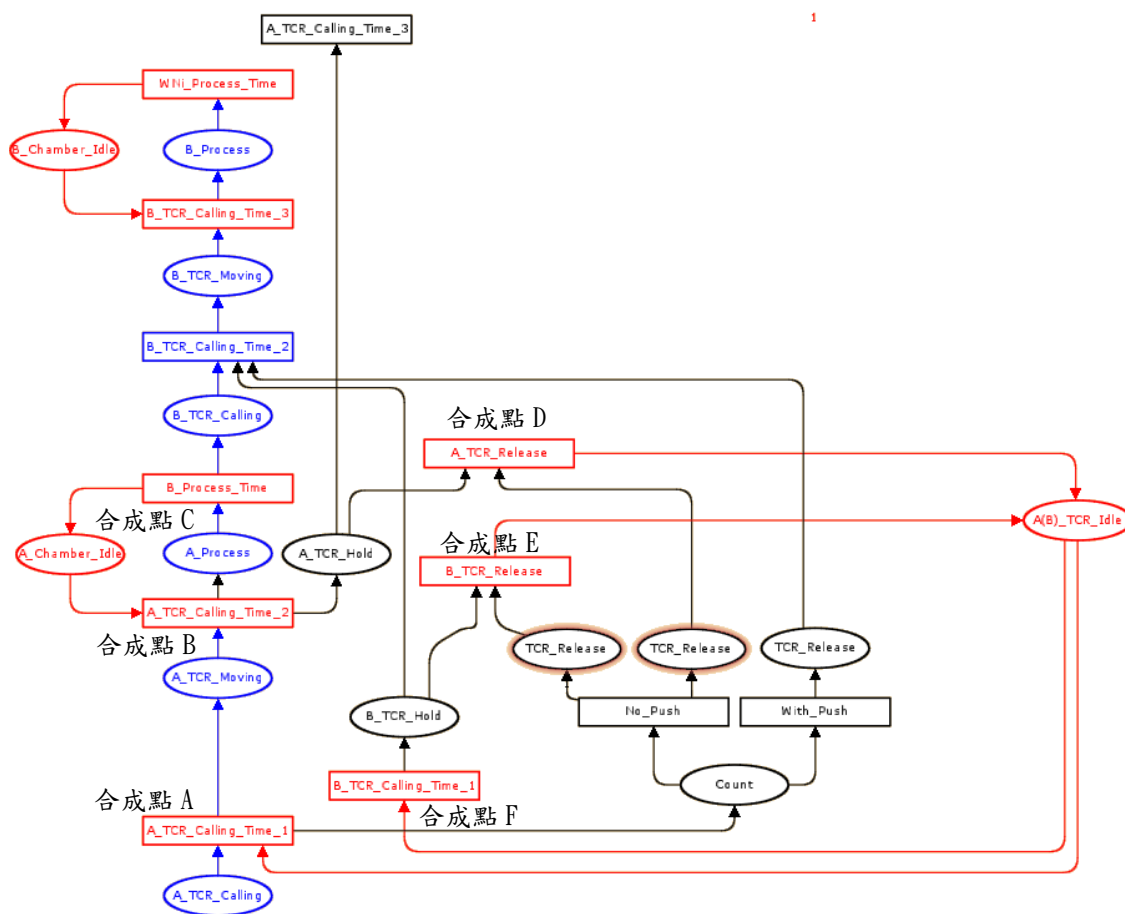
#### 4. 晶圓生產流程模組

$$WP_f = \{P_{WP_f}, T_{WP_f}, B_{WP_f}^-, B_{WP_f}^+, \Theta_{WP_f}, M_{WP_f,0}\}$$



輸出

#### 1. 合成完畢之斐式網 $N = \{P_N, T_N, B_N^-, B_N^+, \Theta_N, M_{N0}\}$



Step1. 合成 有使用手臂控制邏輯之資源模組 與 手臂控制邏輯模組

```

For each transition of CRMs {           //TCRMs = {tCRMsv', v'=1,2,...,|TCRMs|}

    For each transition of CMx {       //TCMx = {tCMxv, v=1,2,...,|TCMx|}

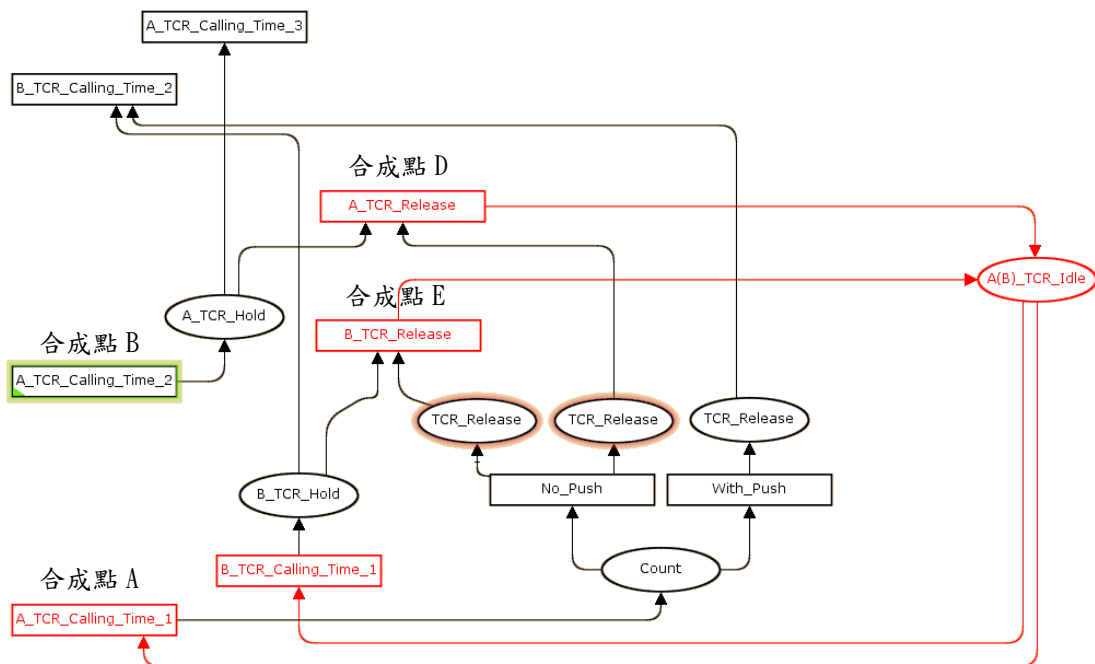
        If TCRMs ∩ TCMx ≠ ∅ then

            N1 = CRMs || CMx       // 表示有 CRMs有使用到 CMx

        End if

    };

};
    
```



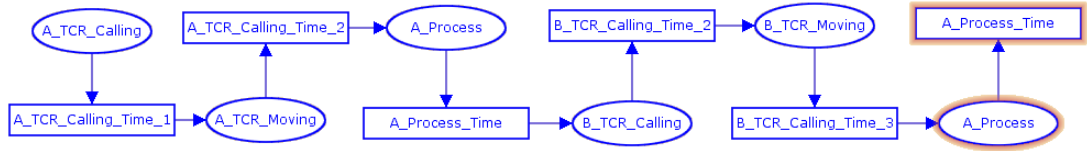
Step2. 合成晶圓生產流程模組

```

For each transition of WPf{           //TWPf = {tWPfv', v'=1,2,...,|TWPf|}
    
```

$N = N \parallel WP_f$

};



Step3. 合成 未使用控制邏輯之資源模組 與 Step2 產出之晶圓生產流程 N

For each transition of N { //  $T_N = \{t_{Nv}, v=1,2,\dots, |T_N|\}$

For each transition of  $RM_j$  { //  $T_{RM_j} = \{t_{RM_jv}, v=1,2,\dots, |T_{RM_j}|\}$

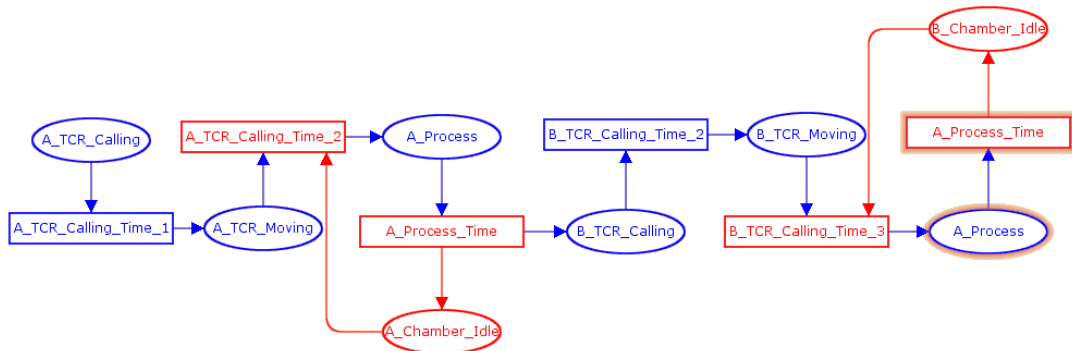
If  $T_N \cap T_{RM_j} \neq \Phi$  then

$N = N \parallel RM_j$  // 表示有 N 有使用到  $RM_j$

End if

};

};



Step4. 合成 Step1 產出之已含控制邏輯之資源模組 N1 與 Step3 產出之 N

For each transition of N { //  $T_N = \{t_{Nv}, v=1,2,\dots, |T_N|\}$



For each transition of  $N_1$  { //  $T_{N_1} = \{t_{N_1^v}, v = 1, 2, \dots, |T_{N_1}|\}$

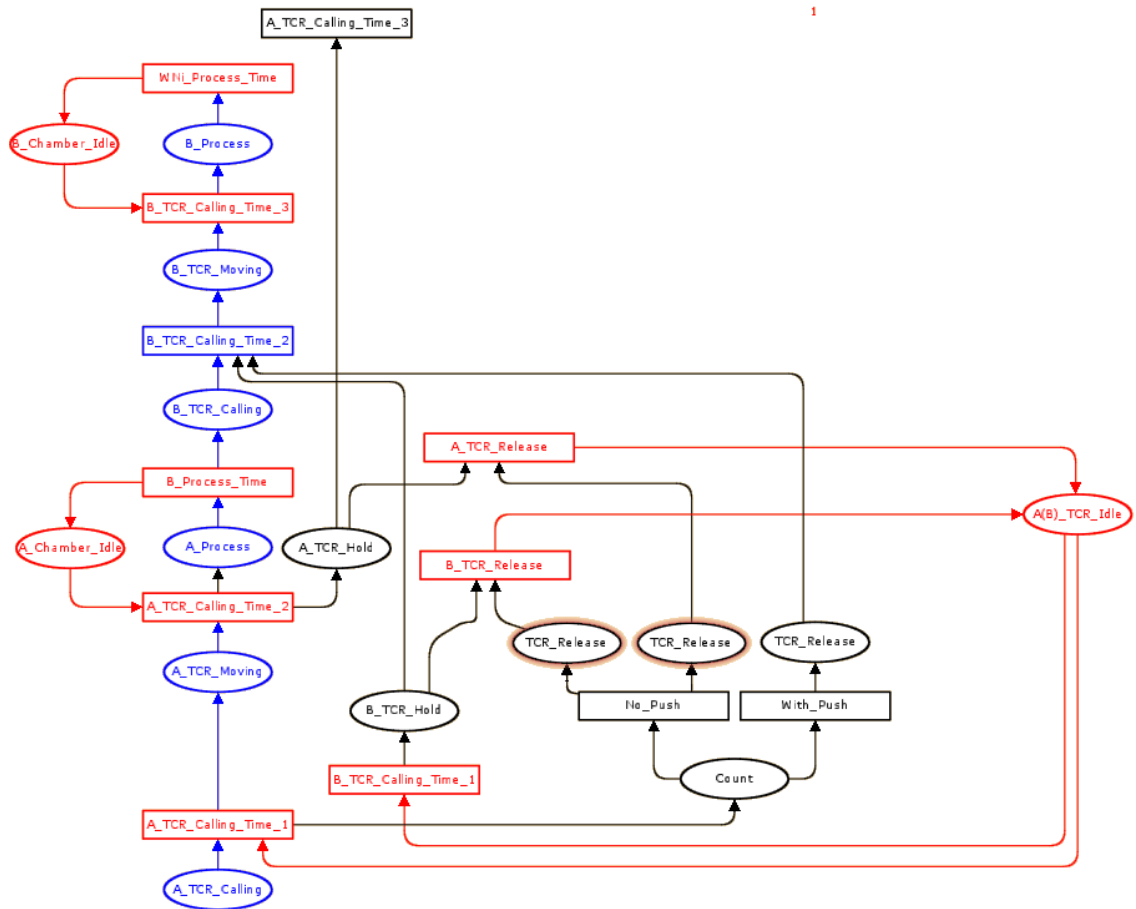
If  $T_N \cap T_{N_1} \neq \Phi$  then

$N = N || N_1$  // 表示有 N 有使用到  $N_1$

End if

};

};



Step5. 完成合成

$N = \{P_N, T_N, B_N^-, B_N^+, \Theta_N, M_{N_0}\}$  即合并之所得

## 4.2 模擬執行與初步驗證

根據合成完畢之叢聚式 PVD 設備雙式網，使用 CPN tools® 開放式軟體進行模擬(如圖 4.2 所示)，我們將其與使用 Push 推式手臂控制邏輯之實際線上機台監測參數進行比較。觀察其手臂控制邏輯影響下，平均晶圓輸出時間間隔之預測誤差值在一標準差之內(如圖 4.3 所示)，在分析手臂控制邏輯運作下手臂隨著時間變化的行為預測情形與實際手臂行為相吻合(如圖 4.4 所示)，最後分析其資源占用情形百分比，與實際線上作業資料也不謀而合。

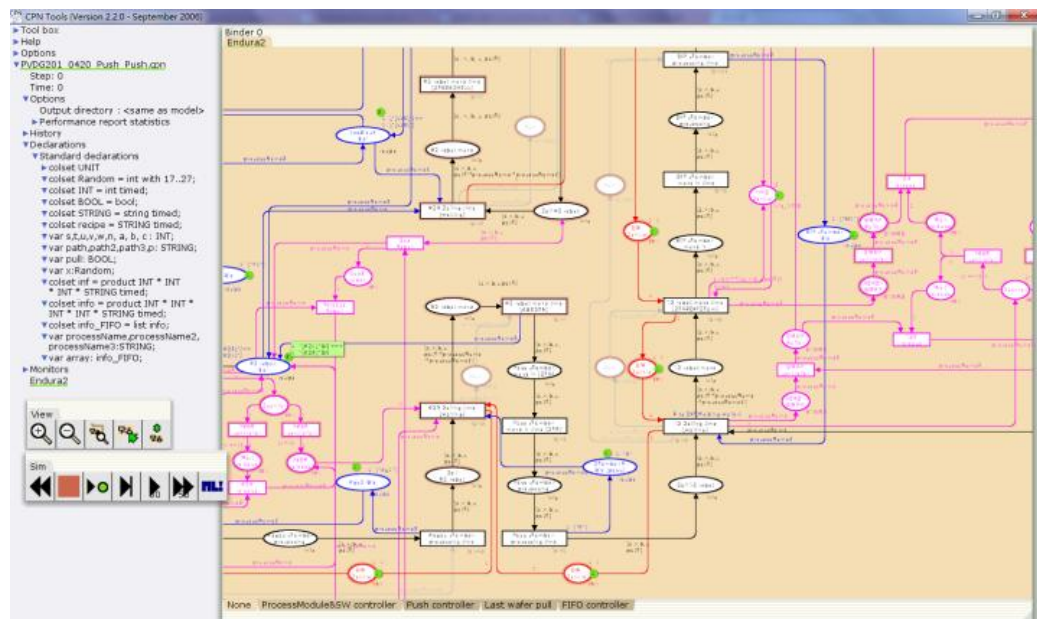


圖 4.2 CPN tools® 模擬介面

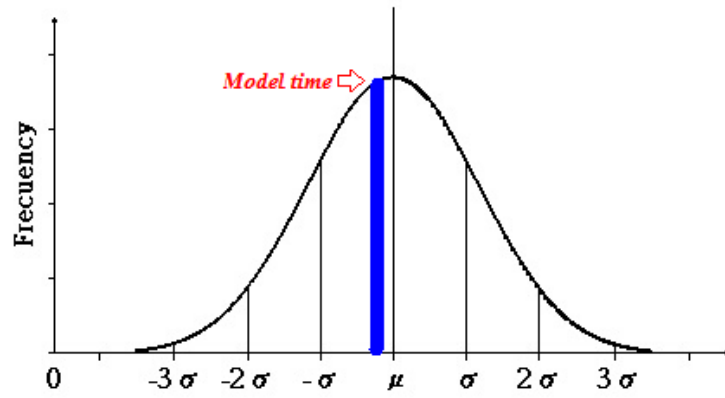


圖 4.3 平均晶圓輸出時間間隔

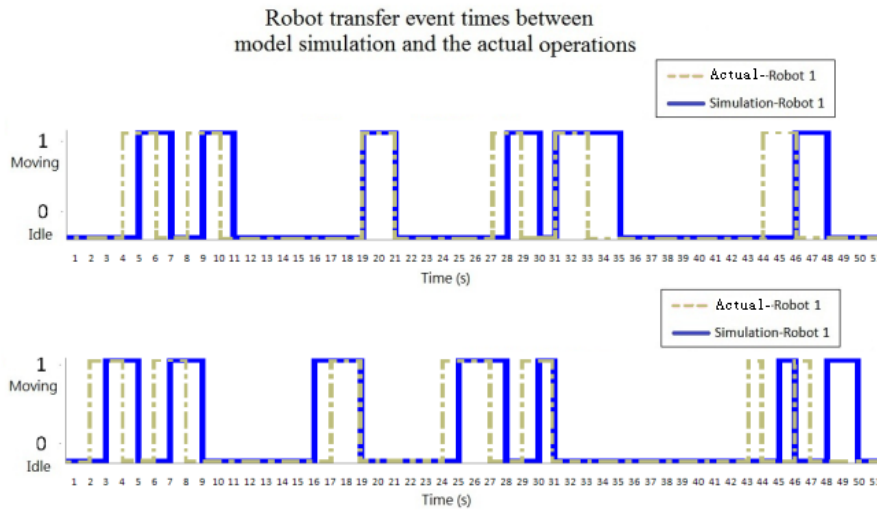


圖 4.4 手臂隨著時間變化的行為預測(1)

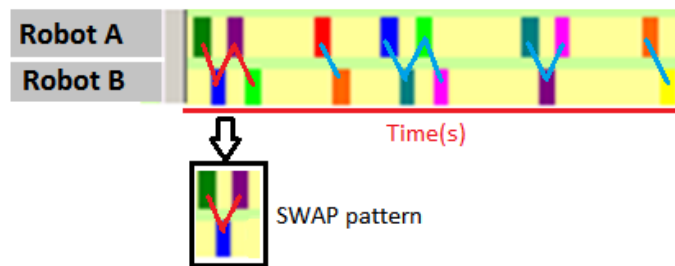


圖 4.5 手臂隨著時間變化的行為預測(2)

## 第五章

### 結論

本研究針對觀察到的叢聚式設備手臂控制邏輯提出一建模流程，此建模流程針對叢聚式設備之作業順序與手臂控制邏輯的彩色時間斐式網建模及其模組成，並以 PVD 設備作為本研究建模討論案例，最終目的是為了建構一個標準化的建模流程。在 PVD 設備作業上可分為 Loadlock、晶圓 process flow 及 Transfer 模組，手臂控制邏輯上可分為 Push 推式、Pull 拉式、FIFO 先進先出式控制邏輯模組，而主要所面對的挑戰為：(1) 如何將觀察到的叢聚式設備手臂控制邏輯建模，及(2) 如何將斐式網模組成。

對此，本研究採用語意式的手臂控制邏輯建模概要建構模組化的彩色時間斐式網，並提出模組化的彩色時間斐式網合成演算法結合 PVD 設備作業模組、手臂控制邏輯模組，最後使用斐式網模擬環境 CPN tools® 進行模擬，針對合成完成之彩色時間斐式網比對實際作業機台進行效能分析，探討 (1)晶圓產出時間間隔 (2) 手臂運輸觸發時間 及(3) 設備占用時間。

根據合成完畢之叢聚式 PVD 設備斐式網，使用 CPN tools® 開放式軟體進行模擬，我們將其與使用 Push 推式手臂控制邏輯之實際線上機台監測參數進行比較。觀察其手臂控制邏輯影響下，平均晶圓輸出時間間隔之預測誤差值在一標準差之內，在分析手臂控制邏輯運作下手臂隨著時間變化的行為預測情形與實際手臂行為相吻合，最後分析其資源占用情形百分比，與實際線上作業資料也不謀而合。

最後我們整理本研究中之建模流程如下圖 5.1 所示，在標準化的建模流程中，我們提出

- (A) 模組化的斐式網建模方式，含括了控制邏輯模組、Loadlock 模組、晶圓生產流程模組 及 一般資源模組。

(B) 設計三步驟模組成演算法將模組化的斐式網合成為一能忠實反映實際線上作業機台之斐式網。

(C) 透過模擬結果轉換機制，我們把 CPN tools 模擬結果透過製程分析軟體來加以分析模型的平均晶圓輸出時間間隔、反應室及手臂資源使用率及特殊控制邏輯下手臂特色呈現結果驗證。

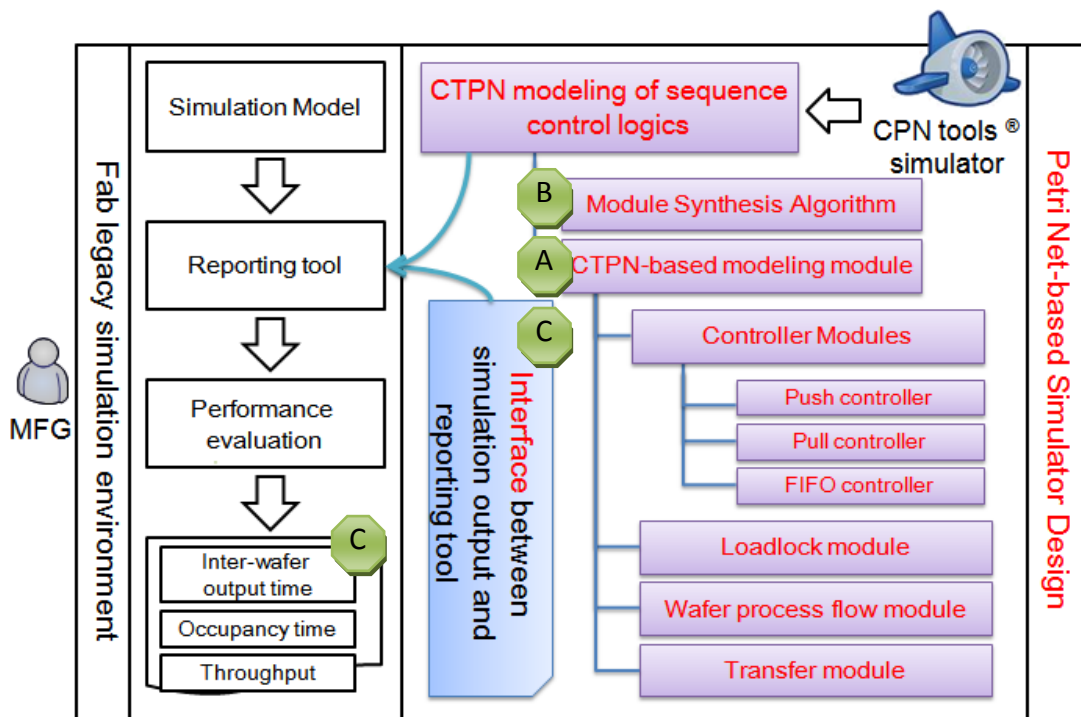


圖 5.1 本研究中之建模流程

本研究著重在於探討如何將叢聚式半導體生產設備之手臂控制邏輯使用斐式網加以建模並進行其效能分析，未來仍有許多後續研究值得後進繼續延伸開花結果：

2. 對於合成完畢之斐式網，我們該如何檢查其之完整、合理性？
3. 完成斐式網建構後，該如何將既有斐式網轉換成相對應之數學模式？

4. 該如何建該完整的手臂控制邏輯語意網，含括現今半導體設備製造業中各種手臂控制邏輯？

## 參考文獻

- [1] G.-M. Chen. (2008). The trend of development of semiconductor technology [Online]. Available: <http://ctle.cyu.edu.tw/tew/Teacher/data/971223/971223-1.pdf>
- [2] T. Pomorski, "Managing Overall Equipment Effectiveness [OEE] to Optimize Factory Performance," Proc. of 1997 IEEE Intl. Symp. on Semi. Mfg. Conf., Oct. S.F., A33 - A36.
- [3] R. S. Srinivasan, "Modeling and Performance Analysis of Cluster Tools Using Petri Nets," IEEE Trans. on Semi. Mfg., 11:394-403, 1998.
- [4] A.N. Swe, A.K. Gupta, A.I. Sivakumar, P. Lendermann, "Cycle Time Reduction at Cluster Tool in Semiconductor 晶圓 Fabrication," Electr. Packaging Tech. Conf. EPTC '06. 8th, SG pp. 671-677 Dec., 2006.
- [5] H. Y. Lee, T. E. Lee, "Scheduling Single-armed Cluster Tools with Reentrant 晶圓 Flows," IEEE Trans. on Semi. Mfg., 19:226-240, 2006.
- [6] CPN tools [Online]. Available: <http://wiki.daimi.au.dk/cpntools>
- [7] Dümmler, M. 2004. "Modeling and Optimization of Cluster Tools in Semiconductor Manufacturing." PhD thesis, Department of Computer Science. University of Würzburg, Germany.
- [8] N. Wu and M. C. Zhou, "Schedulability and scheduling of dual-arm cluster tools with residency time constraints based on Petri net," in Proc. IEEE Conf. Autom. Sci. Eng., Shanghai, China, 2006, pp. 85-90.
- [9] Applied Endura® 2 [Online]. Available: [http://www.appliedmaterials.com/products/assets/conductor\\_deposition/31\\_End2\\_noK\\_PI.pdf](http://www.appliedmaterials.com/products/assets/conductor_deposition/31_End2_noK_PI.pdf)
- [10] F. S. Hsieh and S. C. Chang, "Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems, IEEE Trans. Robotics Automat., vol. 10, no. 2, pp. 196-209, 1994.
- [11] Centura 5200 Systems Platform Functional Description, Applied material Corporation, Taiwan, Oct. 1997.
- [12] E. Keller, J. Bukhman, S. Gonzales, C. Magnella, J. Nulman, R. Mosely, H. Grunes, and A. Tepman, "Manufacturing Evaluation of an Endura 5500 PVD Multichamber System," *Solid state technology*, vol. 35, no2, pp. 71-74, Feb. 1992

## 附錄

### 一、合成演算法對應程式碼

輸入

1. 子斐式網路  $ANet = \{ P_A, T_A, B_A^-, B_A^+, \Theta_A, M_{A0} \}$
2. 子斐式網路  $BNet = \{ P_B, T_B, B_B^-, B_B^+, \Theta_B, M_{B0} \}$

輸出

1. 合成完之斐式網路  $CNet = \{ P_C, T_C, B_C^-, B_C^+, \Theta_C, C_0 \}$

Step1 搜尋兩子斐式網之共同 Transition

If  $T_A \cap T_B \neq \{\Phi\}$  then

Go to step2

Else

Go to step4



End if

```

183:
184: '兩子斐式網合成演算法 (有控制邏輯資源模組 || 手臂控制邏輯模組)
185: Dim _cl_module_array_count As Integer '控制邏輯模組的搜尋計數器
186: Dim _cl_resource_module_array_count As Integer '有控制邏輯資源模組的搜尋計數器
187: For _cl_resource_module_array_count = 0 To _cl_resource_module_array.Count - 1 '針對每個有控制邏輯資源的模組
188:     For _cl_module_array_count = 0 To _cl_module_array.Count - 1 '搜尋有相同Transition naming的手臂控制邏輯模組
189:         Dim i, j As Integer
190:         Dim intersectionCheck As Boolean = False '預設兩子斐式網間Transition沒有交集
191:         For i = 0 To _cl_resource_module_array(_cl_resource_module_array_count).count - 1
192:             For j = 0 To _cl_module_array(_cl_module_array_count).count - 1
193:                 If _cl_resource_module_array(_cl_resource_module_array_count)(i)(3).Equals(_cl_module_array(_cl_module_array_count)(j)(3)) = True Then
194:                     _resource_module_array.Add(Merge(_cl_resource_module_array(_cl_resource_module_array_count), _cl_module_array(_cl_module_array_count)))
195:                     'Algorithm Step1 中 Go to step2 (就是Merge function)
196:                     intersectionCheck = True
197:                 Exit For
198:             '合成兩斐式網, 並儲存至無控制邏輯資源模組集合中
199:             Else
200:                 'Algorithm Step1 中 Go to step4 (就是結束)
201:             End If
202:         Next
203:         If intersectionCheck = True Then Exit For
204:     Next
205:     If intersectionCheck = True Then Exit For
206: Next
207: Next
208: '結束兩子斐式網合成演算法 (有控制邏輯資源 || 手臂控制邏輯)

```

Step2 合成兩子斐式網

Update six-tuples of Petri Net

A	$T_N = T_A \cup T_B \quad // \quad T_A = \{t_{Av}, v'=1,2,\dots,  T_A \}$ $= \{T_A \cap \overline{T_B}, T_A \cap T_B, \overline{T_A} \cap T_B\} \quad // \quad T_B = \{t_{Bv'}, v''=1,2,\dots,  T_B \}$ $= \{t_{Nv}, v=1,2,\dots,  T_A \cap \overline{T_B} ,  T_A \cap \overline{T_B} +1,\dots,  T_A \cap T_B ,  T_A \cap T_B +1, \dots,  T_A \cap \overline{T_B} \}$
B	$P_N = P_A \cup P_B \quad // \quad P_A = \{p_{Au}, u'=1,2,\dots,  P_A \}$ $= \{P_A \cap \overline{P_B}, P_A \cap P_B, \overline{P_A} \cap P_B\} \quad // \quad P_B = \{p_{Bu'}, u''=1,2,\dots,  P_B \}$ $= \{p_{Nu}, u=1,2,\dots,  P_A \cap \overline{P_B} ,  P_A \cap \overline{P_B} +1,\dots,  P_A \cap P_B ,  P_A \cap P_B +1, \dots,  P_A \cap \overline{P_B} \}$
	$\text{New } B_N^- = [0]_{ P_N  \times  T_N } \quad // \text{初始化 } B_N^-$ $\text{New } B_N^+ = [0]_{ P_N  \times  T_N } \quad // \text{初始化 } B_N^+$ <p>For each <math>p_{Nv} \in P_N</math> { <span style="float: right;">// 針對 <math>P_N</math> 中每個元素更新 Marking</span></p>

```

If  $p_{Nu} = p_{Au}$  and  $p_{Nu} = p_{Bu}$ , then //  $P_{Nu}$  同時存在  $P_A$  與  $P_B$  中 (交集)

     $m_{Nu0} = m_{Au0} + m_{Bu0}$  // Place 中的 Marking 相加

     $\mathbf{b}_{Nu}^- = \mathbf{b}_{Au}^-$  // input arc 列向量

     $\mathbf{b}_{Nu}^+ = \mathbf{b}_{Au}^+$  // output arc 列向量

Else if  $p_{Nu} = p_{Au}$ , then //  $P_{Nu}$  存在  $P_A$  (非交集)

     $m_{Nu0} = m_{Au0}$  // 取  $M_{A0}$  中 Place 的初始 Marking

     $\mathbf{b}_{Nu}^- = \mathbf{b}_{Au}^-$  // 將  $\mathbf{b}_{Au}^-$  填入初始化之  $\mathbf{B}_N^-$ 

     $\mathbf{b}_{Nu}^+ = \mathbf{b}_{Au}^+$  // 將  $\mathbf{b}_{Au}^+$  填入初始化之  $\mathbf{B}_N^+$ 

Else //  $P_{Nu}$  存在  $P_B$  (非交集)

     $m_{Nu0} = m_{Bu0}$  // 取  $M_{B0}$  中 Place 的初始 Marking

     $\mathbf{b}_{Nu}^- = \mathbf{b}_{Bu}^-$  // 將  $\mathbf{b}_{Bu}^-$  填入初始化之  $\mathbf{B}_N^-$ 

     $\mathbf{b}_{Nu}^+ = \mathbf{b}_{Bu}^+$  // 將  $\mathbf{b}_{Bu}^+$  填入初始化之  $\mathbf{B}_N^+$ 

}

For each  $t_{Nv} \in T_N$  { // 針對  $T_N$  中每個元素更新時間延遲

    If  $t_{Nv} = t_{Av}$  and  $t_{Nv} = t_{Bv}$ , then { //  $t_{Nv}$  同時存在  $T_A$  與  $T_B$  中 (交集)

         $\theta_{Nv} = \theta_{Bv}$  // 取 Transition 時間延遲的最大值

    Else if  $t_{Nv} = t_{Av}$  //  $t_{Nv}$  僅存在  $T_A$  中 (非交集)

         $\theta_{Nv} = \theta_{Av}$  // 取  $T_A$  中 Transition 時間延遲

    Else //  $t_{Nv}$  僅存在  $T_B$  中 (非交集)

         $\theta_{Nv} = \theta_{Bv}$  // 取  $T_B$  中 Transition 時間延遲

    };

};

```

般)

(宣)

```
478 | Function CreatArcTag ...
516 | Function Merge(ByVal ANet As Object, ByVal BNet As Object)
517 |     Dim CNet As New ArrayList
518 |     Dim x, y As Integer '用來把A,BNet的元素一一的加入CNet當中 (取聯集)
519 |     For x = 0 To ANet.count - 1
520 |         CNet.Add(ANet(x))
521 |     Next
522 |     For y = 0 To BNet.count - 1
523 |         CNet.Add(BNet(y))
524 |     Next
525 |     Dim i, j, m As Integer
526 |     For i = 0 To ANet.count - 1
527 |         For j = 0 To BNet.count - 1
528 |             If ANet(i)(3).Equals(BNet(j)(3)) = True Then '有相同Transition (交集)
529 |                 Dim GoalTID As String = ANet(i)(2)
530 |                 Dim GoalName As String = ANet(i)(3)
531 |                 For m = 0 To CNet.Count - 1
532 |                     If CNet(m)(3).Equals(GoalName) = True Then CNet(m)(2) = GoalTID
533 |                 Next
534 |             End If
535 |         Next
536 |     Next
537 |     Return CNet
538 | End Function
```

A

C

Place 丟回再處理(取聯集)

```
229 | Dim k, t As Integer
230 | Dim comparePlaceID As New ArrayList
231 | Dim check As Boolean = False
232 | '產生Resource的Place
233 | For k = 0 To _resource_module_array(2).Count - 1
234 |     If _resource_module_array(2)(k)(0).Equals("PtoT") = True Then
235 |         For t = 0 To comparePlaceID.Count - 1
236 |             If _resource_module_array(2)(k)(4).Equals(comparePlaceID(t)) = True Then
237 |                 check = True
238 |                 Exit For
239 |             End If
240 |         Next
241 |         If check = False Then
242 |             objXmiTw = CreatPlaceTag(objXmiTw, _resource_module_array(2)(k)(4), _resource_module_array(2)(k)(5))
243 |             comparePlaceID.Add(_resource_module_array(2)(k)(4))
244 |         End If
245 |     End If
246 | End If
247 | check = False
248 | Next
249 | '結束產生Resource的Place
250 |
```

B