

積體電路下線系統的設計與實作：以實體化視觀表重構

葉宸郡¹、李之中²

¹ 中華大學資訊管理學系碩士在職專班研究生。

² 中華大學高階資料庫管理碩士在職學位學程副教授(leecc@chu.edu.tw)

摘要

積體電路從設計到量產所需耗費的時程為積體電路設計公司營運成功的關鍵工作，此段工作作業時程越短，積體電路設計公司就愈有機會獲得競爭優勢。因此，積體電路設計公司需要定期追蹤積體電路的設計開發、積體電路製作生產，推動量產的作業情形。積體電路下線系統整合公司內部所有可用資源，串聯眾多資訊系統，獲得邏輯和實體設計、下線、光罩和晶圓的製作、晶圓針測與封裝等多項工作的作業情形，甚至量產後的各項統計等。積體電路下線系統實際為一個整合資訊系統，在實務應用中執行效能為其重要議題。本研究運用實體化視觀表重構積體電路下線系統，並完成系統設計，以及於工作場域的實作。實證顯示，在系統重構後針對特定下線資料查詢的回應時間僅為重構前的10.8%，使用實體化視觀表重構系統的效益明顯。

關鍵字：積體電路、下線系統、實體化視觀表。

ABSTRACT

In an IC design process, the most important activities are those between the stage of integrated circuit design and stage of production. In the industry, executives and project managers need to monitor the process and the status in the IC design and the production stage. The integrated circuit tape-out system need to integrate all available resources within the company to achieve the expected function. To reach this goal, the integrated circuit tape-out system may integrate heterogeneous information systems, such as logic and entity design process, tape-out process, production status of mask and wafer, wafer probe and packaging process, even the delivery statistics after mass production. Researches proposed that the use of materialize views can enhance the query performance in the distributed computing environments. In this study, the integrated circuit tape-out system is divided into the monitor and the receiver. The materialize views are deployed on the monitor and the receiver. The result shows that the query response time in the integrated circuit tape-out system with materialize views is shorter than that without materialize views.

KEYWORDS: Integrated Circuit, Tape-out, Materialized View.

1 前言

台灣的半導體產業以晶圓代工業為核心，已經發展成垂直整合與分工合作之產業結構。半導體產業由上游至下游依序為積體電路(Integrated Circuit, IC)設計、IC 製造、IC 封裝、IC 測試、IC 通路商、系統廠等[1]。IC 設計業身處在半導體產業鏈的上游，現今消費性電子產品更迭快速，而產品生命週期則日益減短。提供消費性電子產品的 IC 設計業，也在此一快速變動的環境中，面臨巨大的競爭壓力。因此，如何控制新產品開發週期，如何縮短設計與生產的進度，成為營運的重要課題。

積體電路下線系統(IC Tape-out System)[2, 3, 4]的建置，使得 IC 設計公司能夠在新產品開始進行開發時，便能掌控每一個設計與生產作業的進度，縮短新產品的開發時程，甚至上市時間，以此在 IC 設計產業中生存甚或脫穎而出。不僅於此，透過串接每個部門的資訊系統獲得積體電路的設計開發與各階段生產製作的資訊，則可進一步實現跨部門的整合，獲得企業競爭優勢。

積體電路下線系統是一個可以提供積體電路設計開發到量產完成各階段下線資料的整合性資訊系統，涵蓋設計、光罩、晶圓、針測、封裝、最終測試作業等多項工作的作業情形資料[5]。在企業中這些下線資料可能分別儲存於不同廠區遠端資訊系統的資料庫中，如，工程研發(ERD)、企業簽核管理(BPM)、各廠區企業資源規劃(ERP)、目標分析與優化(TANGO)、人力資源(HR)資料庫等。為了取得各階段的下線資料，使用者(高階主管)於應用系統端執行應用程式時，必須至遠端資訊系統資料庫中擷取下線資料。當資料擷取受到網路傳輸或是遠端資訊系統資料庫查詢處理的影響時，下線資料查詢的回應時間將難以預期，最終導致查詢回應時間無法符合使用者期待，遑論提供滿意的使用者經驗。

本研究為了促進積體電路下線系統查詢效能，在不考慮大規模修改現有積體電路下線系統的前提下，嘗試使用實體化視觀表(Materialized View, MV)[6]重構現有下線系統，以改善現有下線系統的查詢效能。系統重構的核心想法為讓積體電路下線系統在近端應用系統擷取遠端資訊系統的下線資料時，能夠直接讀取已經儲存於近端應用系統資料庫中實體化視觀表的複本下線資料，取代直接讀取(Direct Access)遠端資訊系統的資料庫，藉此提升下線系統的查詢效能，同時提供符合期待的使用者體驗，促進下線系統的使用，達到支援企業營運的目的。

本論文後續內容如下，在第二節中報告完成本研究所需的學理與技術的文獻探討，應用現行學理與技術進行的下線系統重構作業的設計與工作場域實作則分別敘明於第三與第四節。第五節為個案討論主要報告於工作場域實作系統後，系統效能的改善結果。第六節則為本論文的總結。

2 文獻回顧

2.1 積體電路下線系統

學者 Mouli 與 Winstead [2] 提出一個將晶圓廠製造執行系統(MES)的規範和功能擴展到產品下線部分的下線執行系統(Tapeout Execution System, TES)。TES 的模組包含整合組態管理、下線作業工作流程的執行和控制、使用者介面、事件管理、資料分析、下線工具管理器、作業控制。在使用者介面模組中，TES 提供了下線狀態儀表板，以此監控下線作業的工作進度。TES 提供即作業狀態、執行指標，並記錄電子設計自動化(Electronic Design Automation, EDA) 工具的

處理參數。工程師亦可經由 TES 的作業控制介面調整作業參數，亦能立即擷取下線資料，進行分析作業。除此之外，如果下線系統發生執行錯誤，TES 也可立即發出警報通知工程師進行排除作業。學者 Leng, Har 與 Tat [3] 提出以關聯式資料管理技術為工具，進行光罩製作(Mask Tooling)的自動化作業，以此改善以人工作業準備光罩製作文件是否存在錯誤疑慮，以及資料正確與否的問題。自動化光罩製作資料準備系統為光罩下線作業提供了良好的作業效能和正確資料。光罩製作文件準備作業的電子化，也能夠排除光罩製作工單的錯誤，並於資料庫中完整留存下線作業資料，以備後續作業或是智慧製造的深度應用。

2.2 實體化視觀表

視觀表(View)在關聯式資料庫系統中定義於某個或數個基底關聯表的虛擬關聯表[7]。建立視觀表時只需以查詢敘明視觀表與基底關聯之間的映對，毋須自基底關聯表匯入資料。使用者對視觀表進行查詢時，資料庫管理系統首先依據視觀表與基底關聯表的映對，再將此一對視觀表的查詢映對成對基底關聯表的查詢，之後執行基底關聯表查詢，並將結果回傳給使用者。常用的視觀表為行列子集、多關聯表合併、統計總覽等。

而實體化視觀表則是將視觀表實體化(Materialization)之後的關聯表[8]。視觀表之運用具有限制存取、簡化查詢、資料獨立、多樣性呈現等優點，但是最大的缺點則是經此擷取資料的回應時間難以預期。為此資料庫廠商建議將視觀表進行實體化，也就是依據視觀表中的查詢，自基底關聯表中擷取資料，並將擷取資料儲存於視觀表中，這種帶有資料的視觀表即為實體化視觀表。

實體化視觀表相對於虛擬視觀表，顯然具有查詢反應時間較佳的優點，但是因為實體化，因此需要若干儲存空間，同時也需要考慮實體化視觀表中的資料(複本)與基底關聯表中的資料(正本)之間資料狀態差異問題[9, 10]。

在實體化視觀表的運用上，以資料倉儲系統的線上資料分析服務應用(On-Line Analytical Processing, OLAP)與分散式資料庫為主要應用場域。OLAP 可提供使用者依據所指定的維度(Dimension)，對事實(Fact)也就是量值，進行向上彙總(Roll-up)、向下鑽取(Drill-down)、切片(Slice)與切塊(Dice)等運算。在資料量大時，自原始資料(Raw Data)進行這些運算，雖然可以得到最新的結果，但是需要花費相當的處理時間，如將所有運算結果都實體化，可有效解決處理時間無法預期問題。但是此一方式，也衍生實體化後資料儲存，以及與原生資料狀態同步問題。因此，在 OLAP 中的視觀表實體化問題中，著重的是如何在查訊處理時間、資料儲存，以及與原生資料狀態同步間獲得同時符合使用者與管理者期待的平衡 [8, 11, 12, 13]。

業界對實體化視觀表的使用則不僅於 OLAP，實務上，實體化視觀表的運用除了對屬同一資料庫的視觀表與基底關聯表進行實體化之外，也可以運用在分散式資料庫的實體化[6, 9]。在此應用中，經由實體化視觀表的建置，將遠端資料實體化於本地端，查詢處理時間可因此獲得確定。在此應用中，關注焦點除了查詢反應時間之外，還包含遠端與近端資料之間的同步問題。在資料倉儲的領域中，學者 Huang 與 Chen [14] 建議應建置包含來源資料和資料倉儲兩個模組的監測機制，通過監測機制對實體化視觀表的同步更新和資料的維護，保持資料的最新狀態和與來源一致性。實體化視觀表的資料同步維護可在以下三個時間點進行[15]，包含立即維護(Immediate)、延遲維護(Deferred)、週期性維護(Periodic)。而維護實體化視觀表的作業方式，可

以重新計算視觀表、將所有資料來源的資料存放在資料倉儲中，以及漸進式維護(Incremental Maintenance)等三種方式進行。

實體化視觀表無論是運用於資料倉儲或是分散式資料庫，在應用領域的學理與技術上均已成熟。將其運用於積體電路下線系統的重構，應為務實與可靠的作業方式。

3 系統設計

本節將說明以實體化視觀表重構下線系統的系統設計，包含系統架構、模組設計及模組間的運作流程。

3.1 系統架構

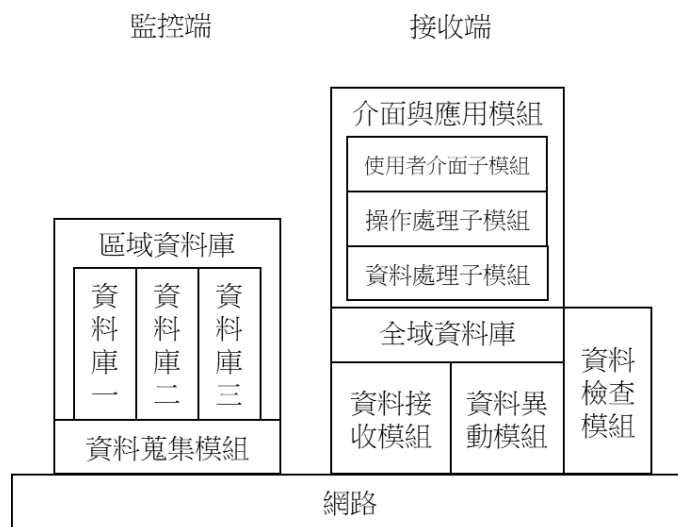


圖 1：系統邏輯架構

本研究之積體電路下線系統邏輯上由位於遠端資訊系統的監控端與近端應用系統的接收端組成，如圖 1 所示。監控端包含區域資料庫(Local Database)(遠端資訊系統資料庫)和資料收集模組；接收端包含全域資料庫、介面與應用、資料接收、資料異動、資料檢查等模組。監控端的主要功能為透過資料收集模組隨時收集區域資料庫(遠端資訊系統資料庫)的異動資訊，再儲存到資料異動紀錄區中。這些異動資訊，再經由網路傳送到接收端的資料接收模組。相對於監控端，接收端的主要功能，是將資料接收模組所接收到的異動資訊儲存到全域資料庫(Global Database)中。當使用者透過使用者介面子模組執行操作，操作處理子模組將分析處理後的需求傳遞給資料處理子模組，資料處理子模組再依需求將全域資料庫的資訊格式化處理後產生報表給使用者介面子模組呈現。資料異動模組主要的功能為當介面與應用模組的資料處理子模組對全域資料庫的資料儲存區作以資料操作語言(Data Manipulation Language, DML)進行異動操作時，將依據所設定的觸發條件對區域資料庫進行操作異動。資料檢查模組主要的功能為檢查資料收集模組、資料接收模組、資料異動模組設定資訊的正確性和資料接收模組的執行紀錄。

3.2 監控端模組設計

位於近端應用系統的監控端包含區域資料庫(遠端資訊系統資料庫)與資料收集模組，同如圖

1 所示，在此說明如下。

1. 區域資料庫(遠端資訊系統資料庫)

積體電路下線系統實際可視為企業內部的資訊整合系統。整合時須對以下資料進行整合，包含研發單位開發積體電路設計進度資料，企業內下線申請簽核表單進度資料，光罩生產的進度資料，晶圓生產的進度資料，針測程式申請簽核表單進度資料，晶圓針測進度和結果資料，封裝申請簽核表單進度資料，封裝生產進度資料，和最後測試的生產進度資料。在本研究中的區域資料庫實際指稱工程研發(Engineering Research Development, ERD)、企業簽核管理(Business Process Management, BPM)，以及企業資源規劃(Enterprise Resource Planning, ERP)等 3 個資訊系統的資料庫。工程研發資料庫記錄積體電路設計、矽智財開發的進度資料；企業簽核管理資料庫記錄內部簽核表單的資訊，如下線申請流程、針測程式申請流程、鐳線封裝核圖申請流程，和專案模塊、表單模塊的資料；企業資源規劃資料庫記錄光罩生產進度模組、晶圓庫存的庫存模組、晶圓的採購進貨模組、晶圓的在製生產進度模組、委外針測的生產進度模組、委外封裝的生產進度模組、委外最終測試的生產進度模組等。

2. 資料收集模組

資料收集模組主要的功能為當區域資料庫發生資料異動時便會將限制值條件資訊、更新時間、新舊值資訊、修改欄位資訊，收集於資料異動紀錄區中，等待接收端進一步的資料更新作業。

3.3 接收端模組設計

接收端共分有全域資料庫、資料接收模組、資料異動模組、資料檢查模組與介面與應用模組等，同如圖 1 所示，分別說明如下。

1. 全域資料庫

全域資料庫由兩個部分組成，第一部分為工程專案管理(Engineering Project Manager, EPM)資料庫，第二部分為來自不同廠區的下線資料。工程專案管理資料庫紀錄系統使用者操作資訊、權限、介面設定，以及整合進度相關資訊。來自不同廠區的下線資料則來自資料接收模組的資料儲存區的區域資料庫異動資訊，為不同廠區多個區域資料庫的複本。

2. 資料接收模組

資料接收模組主要的功能為設定如何接收部署於遠端監控端區域資料庫的異動資訊，以及如何將異動資訊整合(Merge)至全域資料庫的資料儲存區中，如圖 2 所示。

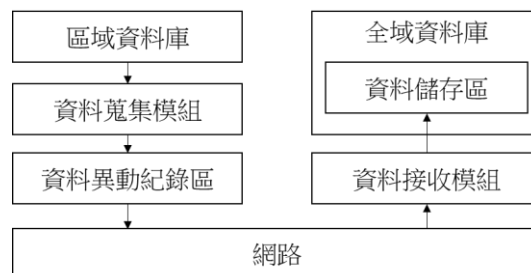


圖 2：資料接收模組

資料接收模組內還需進行資料觸發時機點與資料更新模式兩項設定。

(1) 資料觸發時機點

本項設定的目的為當區域資料庫以交易維護下線資料時，此時下線資料異動資訊是否應該於交易確認之後，立即傳送至接收端地資料接收模組，並更新資料儲存區，或是以批次傳送進行異動資訊傳送，在更新資料儲存區。立即傳送實際即為實體化視觀表對其實體化資料的立即維護；而批次維護則為週期性維護[15]。

(2) 資料更新模式

本項設定的目的為當區域資料庫以交易維護下線資料時，指導資料接收模組如何進行資料儲存區的更新。資料更新模式實際可區分為完全更新、快速更新和強制更新三種方式。快速更新依據來自遠端資料收集模組所提供的資料異動紀錄區的異動資訊進行全域資料庫的資料儲存區更新。完全更新將全域資料庫的資料儲存區資料全部刪除，再重新自遠端資料庫擷取全域資料庫的資料儲存區資料。最後的強制更新則先執行快速更新，若快速更新執行失敗再進行完全更新，以此完成資料儲存區的更新。

3. 資料異動模組

資料異動模組主要的功能為當介面與應用模組的使用者介面子模組呼叫操作處理子模組發出 DML 異動需求時，則資料處理子模組會對全域資料庫的資料儲存區作 DML 操作異動，此時將會經由觸發器所設定的觸發條件對區域資料庫作 DML 操作異動。

4. 資料檢查模組

資料檢查模組主要是檢查資料收集模組、資料接收模組、資料異動模組所設定資訊的正確性，並檢查資料接收模組的執行紀錄。

5. 介面與應用模組

介面與應用模組共分成三個子模組，分別為使用者介面子模組、操作處理子模組、資料處理子模組。使用者介面子模組主要提供給使用者輸入查詢條件並執行查詢命令，和執行新增、刪除、修改指令的網頁介面，用來呼叫操作處理子模組和將接收資料處理子模組產生的結果，然後將產生的執行結果給使用者查閱的網頁介面上。因製作資訊整合平台，所以需於每個查詢網頁的操作介面增加區域的欄位，再將此欄位值傳遞給操作處理子模組。再於結果表格也增加區域欄位，以利使用者區別資料來源為哪個資訊系統的資料。操作處理子模組主要的功能為接收使用者介面子模組的操作行為後，將資訊經過分析處理後再傳遞給資料處理子模組產生結果。因製作資訊整合平台，所以需增加區域參數，來接受使用者介面子模組的區域欄位值，再傳遞到資料處理子模組。資料處理子模組主要的功能為接收操作處理子模組的資訊後，擷取全域資料庫或資料儲存區的資料，再完成資料格式化處理後，將結果回傳給使用者子介面子模組。因製作資訊整合平台，需增加區域的查詢條件，來接收操作處理子模組傳遞來的區域參數。

4 系統實作

為了建構出一套使用實體化視觀表整合企業資訊系統資料庫之積體電路下線系統，本研究在此依據第三節所敘述的系統架構和模組設計進行系統實作。本節依序說明系統實作所使用實

作與開發工具、實作環境，以及監控端與接收端模組的實作作業情形。

4.1 模組實作工具

本實作的各模組所使用到開發工具與程式語言，如表 1 所示。監控端的資料收集模組和接收端的資料接收模組、資料檢查模組、資料異動模組皆使用 Oracle 資料庫的實體化視觀表，並經由 SQL 完成實作。而介面與應用模組則使用 JSP、Java Script、jQuery、Java 等網頁開發前後端語言實作。

表 1：模組實作開發工具和程式語言

監控端/接收端/介面與應用模組	模組名稱	開發工具	實作工具
監控端	資料收集模組	Oracle SQL Developer	SQL
接收端	資料接收模組	Oracle SQL Developer	SQL
接收端	資料檢查模組	Oracle SQL Developer	SQL
接收端	資料異動模組	Oracle SQL Developer	SQL
介面與應用模組	使用者介面子模組	Eclipse	JSP/Java Script/jQuery
介面與應用模組	操作處理子模組	Eclipse	Java/JSON
介面與應用模組	資料處理子模組	Eclipse	Java

4.2 實作環境

本研究使用者介面使用 Java Server Page(JSP)來製作，而介面與應用模組所使用到的 Web 伺服器則採用 Apache Tomcat，全域資料庫和區域資料庫皆為 Oracle 10g 資料庫，監控端和接收端的伺服器皆是採用 Red Hat Enterprise Linux 的作業系統，如圖 3 所示。

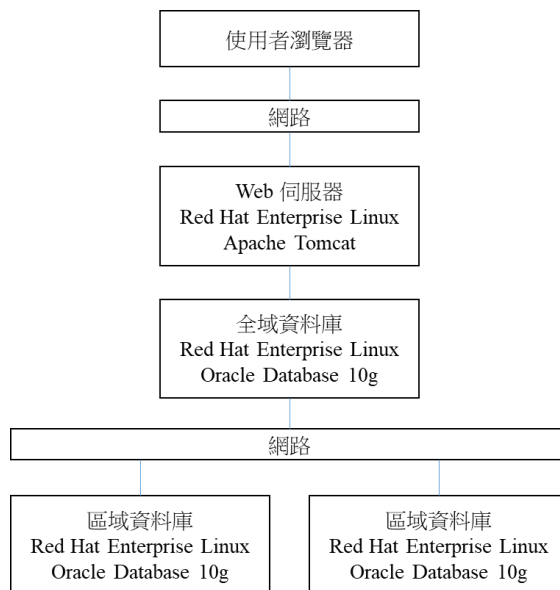


圖 3：系統實體架構

4.3 監控端模組實作

1. 區域資料庫

本研究區域資料庫中，包含工程研發、企業簽核管理，以及企業資源規劃等資料庫，均為於 Red Hat Enterprise Linux 5 作業系統運行的 Oracle 10g 版本資料庫管理系統。

2. 資料收集模組

本模組使用 Oracle 資料庫的實體化視觀表日誌(Materialized View Log)[9, 16]進行區域資料庫異動資料收集作業。作業時，首先對實體化視觀表日誌的資料庫使用者進行二項授權。第一項授權為授予 CREATE TABLE 或 CREATE ANY TABLE 系統權限，第二項授權則為在作業表空間(Tablespace)授予足夠完整儲存實體化視觀表日誌的空間配額。

接著，進行異動資料的限制條件值欄位的設定。位在全域資料庫的實體化視觀表實際上是區域資料庫的複本。當下線系統使用漸進式維護方式[15, 16]進行主複本資料一致性維護時，需要使用限制條件值欄位維持資料主複本之間的同步[9]。實體化視觀表日誌為維持資料主複本一致的資料異動紀錄，必須明確對應資料主複本與資料異動紀錄，除了複本之外，資料異動紀錄也需要指定限制條件值欄位，促使同步主複本資料維護作業的運行。異動資料的限制條件值欄位可以選擇主本資料表的主鍵、ROWID 或是具有唯一性的欄位進行作業。並於建立實體化視觀表日誌時，以 WITH 子句指定限制條件值欄位。

當於區域資料庫完成設定後，實體化視觀表日誌便會產生與主本資料表對應的資料表[9](MLOG\$_TABLENAME, TABLENAME 為對應主本資料表的名稱)於各區域資料庫的資料異動紀錄區中，提供給實體化視觀表要執行快速更新時判斷使用，如圖 4 所示。

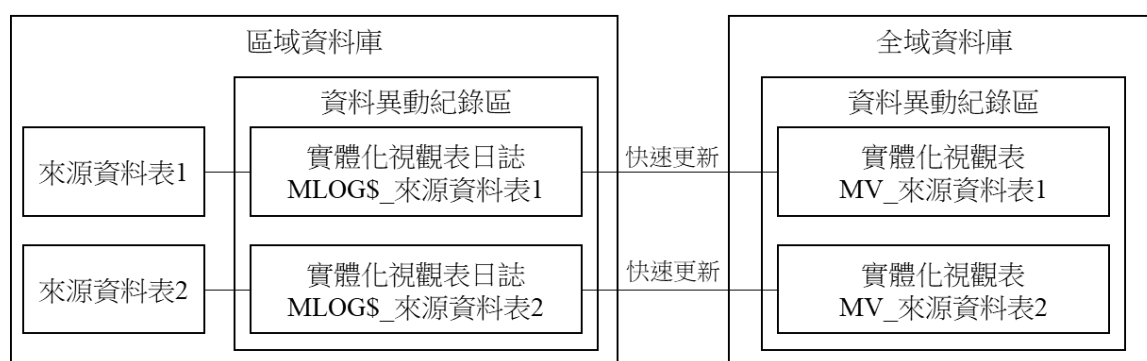


圖 4：實體化視觀表日誌作業方式

4.4 接收端模組實作

1. 全域資料庫

本研究之全域資料庫，包含工程專案管理資料庫，以及來自不同廠區的下線資料，儲存於運行於 Red Hat Enterprise Linux 5 作業系統的 Oracle 10g 資料庫管理系統中。

2. 資料接收模組

資料接收模組接收來自資料異動區的異動資料，並存放於全域資料庫的資料儲存區內。資料儲存區則運用 Oracle 資料庫的實體化視觀表實作。

在進行實體化視觀表建立作業時，須先於全域資料庫中給予資料儲存區使用者 **CREATE ANY MATERIALIZED VIEW** 系統權限。接著建立區域資料庫資料表格與全域資料庫實體化視觀表的主副本資料對照表，同時資料接收模組須於實體化視觀表增加欄位，紀錄下線資料的資料來源，提供給介面與應用模組的資料處理子模組分辨與標示資料來源。

除此之外，建立實體化視觀表時，另需進行資料觸發事件，以及資料更新模式兩項設定，分別說明如下。

(1) 資料觸發事件

資料觸發事件主要是設定實體化視觀表更新事件觸發的條件。此條件如設定為立即更新(**ON COMMIT**)，當區域資料庫的主本資料表有交易確認(**COMMIT**)時，為於全域資料庫資料儲存區的對應實體化視觀表將立即更新。值得注意的是此一作業方式存在若干限制條件，實體化視觀表若使用 **DATABASE LINK** 物件、**UNION ALL** 運算、子查詢物件，或者資料異動使用 **FOR UPDATE** 子句皆無法使用立即更新觸發事件。若條件設定為手動更新(**ON DEMAND**)即便區域資料庫的主本資料表有交易確認，全域資料庫資料儲存區的實體化視觀表不會被立即更新，而是等待接收到 **DBMS_MVIEW.REFRESH** 指令之後才進行實體化視觀表的更新。此一觸發時機方式較無限制條件。實務上，手動更新可配合資料庫管理系統的工作排程作業，自動進行實體化視觀表的更新作業。

(2) 資料更新模式

實體化視觀表的資料更新模式設定主要為設定如何自區域資料庫更新全域資料庫資料儲存區的作業方式。此項作業方式的設定包含快速更新(**FAST**)、完全更新(**COMPLETE**)、強制更新(**FORCE**)等。快速更新實際即為漸增式維護[15, 16]，須配合實體化視觀表日誌，若設定為此一更新模式，當執行於區域資料庫表格的交易確認時，將針對異動資料以實體化視觀表日誌進行全域資料庫資料儲存區實體化視觀表的更新。若設定為完全更新，則先將全域資料庫的資料儲存區實體化視觀表資料刪除後，再重新至區域資料庫擷取並重置實體化視觀表的資料。若設定為強制更新則以先嘗試執行快速更新，若執行失敗再進行完全更新的程序作業。

3. 資料異動模組

資料異動模組主要功能為在應用系統端啟動觸發器(**Trigger**)經由網路更新區域資料庫的下線資料，其作業方式說明如下。首先授予全域資料庫使用者建立觸發器 **CREATE ANY TRIGGER** 系統權限。接著，對資料儲存區的實體化視觀表以 **FOR UPDATE** 子句開啟異動模式，該實體化視觀表便可以 **DML** 進行資料異動操作。最後，於全域資料庫中的實體化視觀表建立觸發器，以此經由 **DATABASE LINK** 更新區域資料庫中下線資料。

4. 資料檢查模組

資料檢查模組主要是檢查資料收集模組、資料接收模組和資料異動模組設定資訊的正確性。本研究使用 Oracle 資料庫所提供的資料字典視觀表(**Data dictionary view**)進行本項檢查，所使用的資料字典視觀表臚列如後。

USER_MVIEW_LOGS、USER_BASE_TAB LE_MVIEWS、USER_SNAPSHOT_LOGS、USER_SEGMENTS、USER_MVIEWS、USER_MVIEW_ANALYSIS、USER_JOBS、USER_REGISTERED_MVIEWS。

5. 介面與應用模組

介面與應用模組主要應用到 Java Web 中的 MVC(Model View Controller)架構，MVC 架構基本概念是把 Model 當作資料，View 則是呈現給使用者的介面，而 Controller 則負責邏輯的處理。所以使用者介面子模組便是 MVC 架構的 View，資料處理子模組和操作處理子模組則是使用了 MVC 的 Model 和 Controller，而在 Model 與 Controller 之間則使用了 Action、Service、Dao 的三層式架構。Action 層主要存放一般的商業邏輯，Service 層則存放需要進行交易的邏輯，Dao 層便只存放資料庫須轉成 Model 物件的邏輯。

5 實作成效

本節以個案方式說明本研究於工作場域時建後的實際成效。個案公司為一消費性電子產品的 IC 設計公司，其資訊部門常接到產品專案經理反應積體電路下線系統的查詢效能不佳，使用者介面操作反應很慢等問題，造成終端使用者(產品專案經理)對下線系統的使用意願低落，下線系統建置未能達成預期效應。同時下線系統在未來尚有數個廠區的遠端資訊系統資料庫需要導入，可以預期處理資料量將更加龐大，若現行系統效能未能改善，查詢效能不佳的問題將更加嚴重。因此，個案公司希望藉由現行下線系統的重構改善現行下線系統的效能。

個案公司經由實體化視觀表重新建構積體電路下線系統後，進行了系統重構前後的查詢效能的差異比較。比較方式為以系統中兩個較常被查詢的視觀表為對象。查詢時皆不設定查詢條件，以全表掃描(Full Scan)方式進行查詢。作業工具使用 Oracle 所提供的 SQL Developer 開發工具。對 *_FAB_WIP 視觀表與*_ERD_ALL 視觀表的作業結果說明如下。

1. *_FAB_WIP 視觀表

對 *_FAB_WIP 視觀表分別於重構前後執行 SELECT * FROM *_FAB_WIP 敘述。重構前的反應時間為 30.068 秒，而重構後的反應時間為 3.26 秒，如圖 5 所示，重構後的反應時間僅是重構前反應時間的 10.8%。

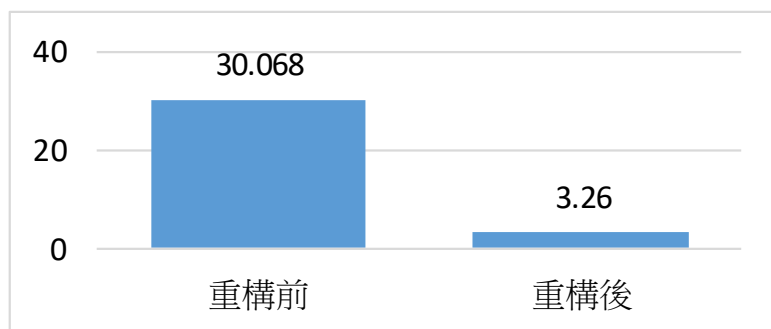


圖 5：*_FAB_WIP 視觀表重構前後反應時間

2. *_ERD_ALL 視觀表

對 *_ERD_ALL 視觀表分別於重構前後執行 `SELECT * FROM *_ERD_ALL` 敘述。重構前的反應時間為 16.027 秒，而重構後的反應時間為 6.037 秒，如圖 6 所示，重構後的反應時間是重構前反應時間的 37.7%。

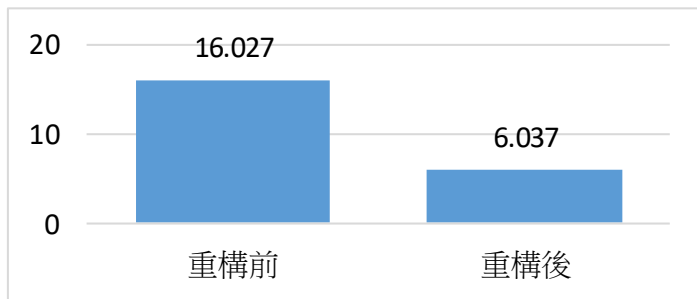


圖 6：*_ERD_ALL 視觀表重構前後反應時間

依據此處對 *_FAB_WIP 與 *_ERD_ALL 視觀表的作業結果，可得知透過實體化視觀表重新建構積體電路下線系統後，系統中兩個較常使用到視觀表的查詢效能確實有所提升。

6 結論

本研究使用實體化視觀表，重構積體電路下線系統。經由實體化視觀表重新建構積體電路下線系統後，因已經將遠端各種資訊系統的資料，集中於全域資料庫中，下線系統執行效能方面比透過網路分別讀取各個資訊系統有所提升。

本研究的實務意涵為 IC 設計公司可透過本技術報告的內容來改善積體電路下線系統的查詢效能，增加使用率。使每個部門願意協助提供積體電路的設計開發與各階段生產製作的進度，實現跨部門的整合能力。使高階主管和專案經理能掌控新產品設計與生產的進度，便能縮短新產品開發時程，降低企業的開發成本，甚至能夠縮短新產品上市的時間，如此才能在競爭激烈的 IC 設計產業環境中脫穎而出。

7 參考文獻

1. 黃千玲 (2018)，從商業生態系統探討半導體產業鏈之競合關係，未出版碩士論文，國立台灣科技大學管理研究所。
2. Mouli, Chandra, Winstead, C. H. (2007). “Tapeout Execution System (TES), a key enabler of DFM/Co-optimization,” in Proc. International Symposium on Semiconductor Manufacturing, USA, 1-4.
3. Ng Chow Leng, Alfonso Jee Chiew Har, Ow Yong Wai Tat (2012). “System management with relational database for mask tape-out.” In Proc. 2012 4th Asia Symposium on Quality Electronic Design (ASQED), 128-132.
4. Park, H., Kim, S., Kim, T., Kim, J., Park, Y., Eom, Y.I. (2015) Resource-Aware Job Scheduling and Management System for Semiconductor Application. In: Park DS., Chao

- HC., Jeong YS., Park J. (eds) *Advances in Computer Science and Ubiquitous Computing. Lecture Notes in Electrical Engineering*, vol 373. Springer, Singapore.
5. 葉宸郡 (2018), 以實體化視觀表建構積體電路下線系統之設計與實作, 未出版碩士論文, 中華大學資訊管理學系碩士在職專班。
 6. Jiang, Y., Zhang, N. and Fang, Y.(2019). "The Analysis and Design of Ship Monitoring System Based on Hybrid Replication Technology," in Proc. 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 456-459.
 7. 曾守正(2010), 資料庫系統理論與實務, 第三版, 華泰文化, 台北。
 8. 林聖斌 (2002), 物件關聯式資料倉儲之實體化視域選擇, 未出版碩士論文, 元智大學資訊管理系碩士班。
 9. Urbano, Randy. (2007), *Materialized View Concepts and Architecture*, Oracle 10g Advanced Replication, Oracle Database Documentation, 10g Release 2 (10.2), Oracle.
 10. 張慶賀 (2003), 資料倉儲中實體化視域自我維護之研究, 未出版碩士論文, 朝陽科技大學資訊管理系碩士班。
 11. Padmaja Potineni (2017), "Basic Materialized Views", *Database Data Warehousing Guide*, Oracle.
 12. 洪明傳 (2006), 有效率的資料倉儲與資料挖掘技術之研究, 博士論文, 逢甲大學資訊工程所。
 13. Kumar, Amit, Kumar, T. V. V. (2017). "Materialized view selection using discrete genetic operators based particle swarm optimization," in Proc. 2017 International Conference on Inventive Systems and Control (ICISC), 1-5.
doi: 10.1109/ICISC.2017.8068675.
 14. Huang, X. and Chen, Q. (2011), "A maintainable model of materialized view based on data warehouse," in Proc. of 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), 1974-1977.
 15. Jogekar, Ravindra N., Mohod, Ashish (2013), "Design and Implementation of Algorithms for Materialized View Selection and Maintenance in Data Warehousing Environment", *International Journal of Emerging Technology and Advanced Engineering*, 3(9), 464-470.
 16. 李皆興 (2002), 物件導向資料倉儲中 SPJ 視域與聚合視域之漸進式維護, 碩士論文, 東吳大學資訊科學系。
 17. Lorentz, Diana (2005). "CREATE MATERIALIZED VIEW LOG", *Database SQL Reference*, Oracle Database Documentation, 10g Release 2 (10.2), Oracle.
Lorentz, Diana (2005). "CREATE MATERIALIZED VIEW", *Database SQL Reference*, Oracle Database Documentation, 10g Release 2 (10.2), Oracle.