



(19)中華民國智慧財產局

(12)發明說明書公開本

(11)公開編號：TW 201413593 A

(43)公開日：中華民國 103 (2014) 年 04 月 01 日

(21)申請案號：102116782

(22)申請日：中華民國 102 (2013) 年 05 月 10 日

(51)Int. Cl. : **G06F9/46 (2006.01)**

(30)優先權：2012/06/15 美國 13/524,921

(71)申請人：萬國商業機器公司(美國) INTERNATIONAL BUSINESS MACHINES CORPORATION (US)

美國

(72)發明人：葛瑞納 丹 F GREINER, DAN F. (US)；傑可比 克理斯俊 JACOBI, CHRISTIAN (DE)；史洛歌 提摩西 J SLEGEL, TIMOTHY J. (US)

(74)代理人：陳長文

申請實體審查：無 申請專利範圍項數：20 項 圖式數：23 共 127 頁

(54)名稱

異動處理

TRANSACTIONAL PROCESSING

(57)摘要

經由一異動開始指令起始一異動。在該異動之執行期間，該異動可中止。若該異動中止，則進行關於異動型別之一判定。基於該異動為一第一異動型別，在該異動開始指令處繼續執行，且基於該異動為一第二型別，在該異動開始指令之後的一指令處繼續(resuming)執行。不管異動型別，繼續執行皆包括恢復該異動開始指令中指定之一或多個暫存器且捨棄異動儲存。對於一異動型別(非受限異動)，該繼續包括將資訊儲存於一異動診斷區塊中。

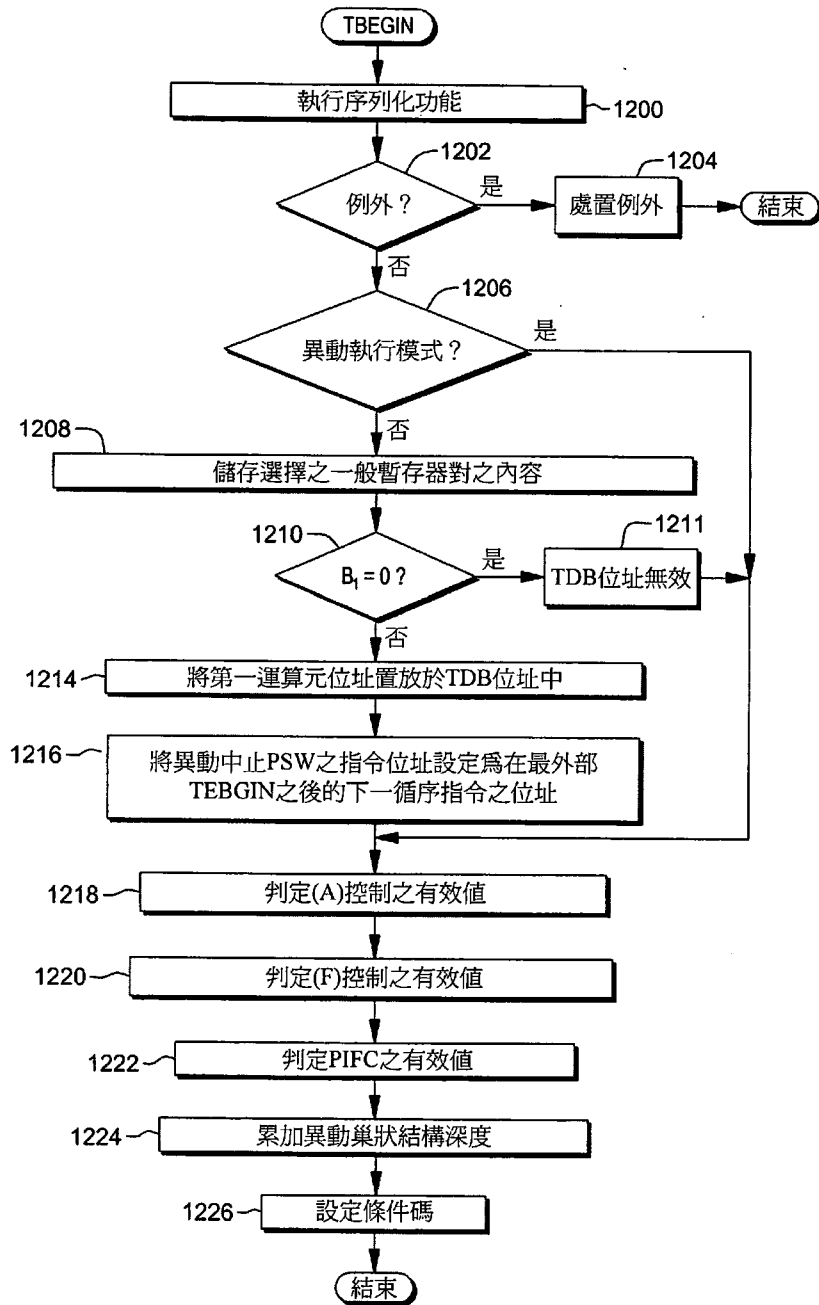


圖12



(19)中華民國智慧財產局

(12)發明說明書公開本

(11)公開編號：TW 201413593 A

(43)公開日：中華民國 103 (2014) 年 04 月 01 日

(21)申請案號：102116782

(22)申請日：中華民國 102 (2013) 年 05 月 10 日

(51)Int. Cl. : **G06F9/46 (2006.01)**

(30)優先權：2012/06/15 美國 13/524,921

(71)申請人：萬國商業機器公司(美國) INTERNATIONAL BUSINESS MACHINES CORPORATION (US)

美國

(72)發明人：葛瑞納 丹 F GREINER, DAN F. (US)；傑可比 克理斯俊 JACOBI, CHRISTIAN (DE)；史洛歌 提摩西 J SLEGEL, TIMOTHY J. (US)

(74)代理人：陳長文

申請實體審查：無 申請專利範圍項數：20 項 圖式數：23 共 127 頁

(54)名稱

異動處理

TRANSACTIONAL PROCESSING

(57)摘要

經由一異動開始指令起始一異動。在該異動之執行期間，該異動可中止。若該異動中止，則進行關於異動型別之一判定。基於該異動為一第一異動型別，在該異動開始指令處繼續執行，且基於該異動為一第二型別，在該異動開始指令之後的一指令處繼續(resuming)執行。不管異動型別，繼續執行皆包括恢復該異動開始指令中指定之一或多個暫存器且捨棄異動儲存。對於一異動型別(非受限異動)，該繼續包括將資訊儲存於一異動診斷區塊中。

發明摘要

※ 申請案號：10211698 ✓

※ 申請日：102.5.10

※IPC 分類：G06F 9/46 (2006.01)

【發明名稱】

異動處理

TRANSACTIONAL PROCESSING

【中文】

- 經由一異動開始指令起始一異動。在該異動之執行期間，該異動可中止。若該異動中止，則進行關於異動型別之一判定。基於該異動為一第一異動型別，在該異動開始指令處繼續執行，且基於該異動為一第二型別，在該異動開始指令之後的一指令處繼續（resuming）執行。不管異動型別，繼續執行皆包括恢復該異動開始指令中指定之一或多個暫存器且捨棄異動儲存。對於一異動型別(非受限異動)，該繼續包括將資訊儲存於一異動診斷區塊中。

【英文】

- A transaction is initiated via a transaction begin instruction. During execution of the transaction, the transaction may abort. If the transaction aborts, a determination is made as to the type of transaction. Based on the transaction being a first type of transaction, resuming execution at the transaction begin instruction, and based on the transaction being a second type, resuming execution at an instruction following the transaction begin instruction. Regardless of transaction type, resuming execution includes restoring one or more registers specified in the transaction begin instruction and discarding transactional stores. For one type of transaction, the unconstrained transaction, the resuming includes storing information in a transaction diagnostic block.

【代表圖】

【本案指定代表圖】：第（12）圖。

【本代表圖之符號簡單說明】：

無

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

（無）

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】

異動處理

TRANSACTIONAL PROCESSING

【技術領域】

一或多項態樣大體上係關於多處理計算環境，且尤其係關於此類計算環境內之異動處理。

【先前技術】

多處理器程式設計中之持久挑戰為由多個中央處理單元(CPU)對同一儲存位置之更新的挑戰。更新儲存位置之許多指令(甚至包括諸如AND之簡單邏輯運算)用對該位置之多次存取來進行該更新。舉例而言，首先提取儲存位置，且接著儲存回已更新結果。

爲了使多個CPU安全地更新同一儲存位置，序列化對該位置之存取。用先前由International Business Machines Corporation提供之S/360架構引入的一個指令(TEST AND SET指令)提供儲存位置之連鎖更新(interlocked update)。連鎖更新意謂：如由其他CPU及輸入/輸出(I/O)子系統(例如，通道子系統)所觀察，指令之整個儲存存取看來像是不可部分完成地發生。稍後，由International Business Machines Corporation提供之S/370架構引入COMPARE AND SWAP及COMPARE DOUBLE AND SWAP指令，該等指令提供執行連鎖更新之較精密方式且允許通常被稱爲鎖定字組(或旗號)之實施的實施。最近新增之指令已提供額外連鎖更新能力，包括COMPARE AND SWAP AND PURGE及COMPARE AND SWAP AND STORE。然而，所有此等指令提供用於僅單一儲存位置之連鎖。

較複雜之程式技術可需要多個儲存位置之連鎖更新，諸如，當將一元素新增至一雙向連結清單(doubly-linked list)時。在此類操作中，前向指標及向後指標兩者皆看來像是被同時地更新，如由其他CPU及I/O子系統所觀察。爲了實現此類多重位置更新，強程式使用分離之單一序列化點，諸如，鎖定字組。然而，鎖定字組可提供比所保證之序列化層級粗略得多的序列化層級；舉例而言，鎖定字組可序列化數百萬個元素之整個佇列，即使僅兩個元素正被更新亦如此。程式可結構化資料以使用較精細粒度級序列化(例如，鎖定點階層)，但彼情形引入額外問題，諸如，在違反該階層時之潛在鎖死情形，及在程式遭遇錯誤同時保持一或多個鎖定期或在不能獲取鎖定期之復原問題。

除了以上內容以外，亦存在程式可執行可或可不引起例外條件之指令序列的眾多情境。若未發生例外條件，則程式繼續；然而，若辨識例外，則程式可採取校正動作以消除例外條件。作爲一實例，Java可在(例如)理論式執行、函式之部分內嵌及/或指標空值檢查之重新定序中利用此類執行。

在諸如由International Business Machines Corporation提供之z/OS及其前身的傳統作業系統環境中，程式建立復原環境以攔截程式可遭遇之任何程式例外條件。若程式未攔截到例外，則作業系統通常針對作業系統未準備好進行處置之例外而異常地終止程式。建立及利用此類環境係昂貴且複雜的。

【發明內容】

經由提供一種用於控制一異動在一計算環境中之執行之電腦程式產品而克服先前技術之缺點且提供優點。該電腦程式產品包括一電腦可讀儲存媒體，該電腦可讀儲存媒體可由一處理電路讀取且儲存指令以供該處理電路執行來執行一方法。舉例而言，該方法包括：由一

處理器經由一異動開始指令在一異動模式下起始一異動之執行，該異動開始指令開始一第一異動型別或一第二異動型別中任一者，該異動包含複數個待執行指令；基於處於異動模式，有效地延遲將異動儲存交付至主記憶體直至一選擇之異動完成為止；基於該異動中止且基於該異動為一第一異動型別，在該異動開始指令處繼續執行；及基於該異動中止且基於該異動為一第二異動型別，在程式次序上緊接在該異動開始指令之後的下一循序指令處繼續執行。

本文亦描述及主張與一或多個實施例相關之方法及系統。另外，本文亦描述且可主張與一或多個實施例相關之服務。

貫穿全文實現額外特徵及優點。其他實施例及態樣在本文中得以詳細地描述且被看作本發明之部分。

【圖式簡單說明】

在本說明書完結時於申請專利範圍中將一或多項態樣特定地指出且清楚地主張為實例。前述及其他目標、特徵及優點自結合隨附圖式而進行之以下詳細描述變得顯而易見，在該等圖式中：

圖1描繪計算環境之一實施例；

圖2A描繪異動開始(TBEGIN)指令之一實例；

圖2B描繪圖2A之TBEGIN指令之欄位的另外細節之一實施例；

圖3A描繪異動開始受限(TBEGINC)指令之一實例；

圖3B描繪圖3A之TBEGINC指令之欄位的另外細節之一實施例；

圖4描繪異動結束(TEND)指令之一實例；

圖5描繪異動中止(TABORT)指令之一實例；

圖6描繪巢狀異動之一實例；

圖7描繪NONTRANSACTIONAL STORE(NTSTG)指令之一實例；

圖8描繪EXTRACT TRANSACTION NESTING DEPTH(ETND)指令之一實例；

圖9描繪異動診斷區塊之一實例；

圖10描繪中止之實例原因，連同關聯中止碼及條件碼；

圖11描繪與執行TBEGINC指令相關聯之邏輯之一實施例；

圖12描繪與執行TBEGIN指令相關聯之邏輯之一實施例；

圖13描繪與執行TEND指令相關聯之邏輯之一實施例；

圖14描繪與異動中止處理相關聯之邏輯之一實施例；

圖15A描繪與異動處理相關聯之邏輯之一實施例；

圖15B描繪與在非受限異動之中止之後繼續處理相關聯之邏輯之一實施例；

圖16A至圖16B描繪將佇列元素插入至佇列元素雙向連結清單中之實例；

圖17描繪電腦程式產品之一實施例；

圖18描繪主機電腦系統之一實施例；

圖19描繪電腦系統之另外實例；

圖20描繪包含電腦網路之電腦系統之另一實例；

圖21描繪電腦系統之各種元件之一實施例；

圖22A描繪圖21之電腦系統之執行單元之一實施例；

圖22B描繪圖21之電腦系統之分支單元之一實施例；

圖22C描繪圖21之電腦系統之載入/儲存單元之一實施例；及

圖23描繪模擬主機電腦系統之一實施例。

【實施方式】

根據一態樣，提供一種異動執行(TX)設施。此設施提供用於指令之異動處理，且在一或多個實施例中提供如下文所描述之不同執行模式，以及異動處理之巢狀層級。

異動執行設施引入被稱作異動執行(TX)模式之CPU狀態。在CPU重設之後，CPU並不處於TX模式。CPU藉由TRANSACTION BEGIN

指令進入TX模式。CPU藉由(a)最外部TRANSACTION END指令(關於內部指令及外部指令之更多細節如下)抑或(b)正被中止之異動而離開TX模式。雖然處於TX模式，但由CPU進行之儲存存取看來像是區塊並行的，如由其他CPU及I/O子系統所觀察。儲存存取(a)在最外部異動在無中止之情況下結束時被交付至儲存體(亦即，例如，在CPU本端之快取記憶體或緩衝器中進行之更新經傳播及儲存於真實記憶體中且可為其他CPU所見)，抑或(b)在異動中止時被捨棄。

異動可為巢狀。亦即，在CPU處於TX模式時，CPU可執行另一TRANSACTION BEGIN指令。使CPU進入TX模式之指令被稱作最外部TRANSACTION BEGIN；相似地，程式據稱處於最外部異動。TRANSACTION BEGIN之後續執行被稱作內部指令；且程式正執行內部異動。模型提供最小巢狀結構深度及模型相依最大巢狀結構深度。EXTRACT TRANSACTION NESTING DEPTH指令傳回當前巢狀結構深度值，且在一另外實施例中可傳回最大巢狀結構深度值。此技術使用被稱作「扁平化巢狀結構(flattened nesting)」之模型，其中任何巢狀結構深度處之中止條件造成異動之所有層級中止，且控制傳回至在最外部TRANSACTION BEGIN之後的指令。

在異動之處理期間，由一個CPU進行之異動存取據稱與(a)由另一CPU進行之異動存取或非異動存取衝突，抑或與(b)由I/O子系統進行之非異動存取衝突(若兩種存取係針對同一快取行內之任何位置，且該等存取中之一者或兩者為儲存)。換言之，為了使異動執行有生產力，CPU不應被觀察到進行異動存取直至異動存取交付為止。此程式設計模型在某些環境中可高度地有效；例如，數百萬個元素之雙向連結清單中之兩個點的更新。然而，若針對正被異動地存取之儲存位置存在許多爭用，則該程式設計模型可較不有效。

在一個異動執行模型(在本文中被稱作非受限異動)中，當異動中

止時，程式可嘗試重新驅動異動以希望不再存在中止條件，或程式可「後援」至等效非異動路徑。在另一異動執行模型(在本文中稱作受限異動)中，已中止異動係由CPU自動地重新驅動；在不存在限制違反時，受限異動確信最終完成。

當起始異動時，程式可指定各種控制，諸如：(a)在異動中止時哪些一般暫存器恢復為其原始內容；(b)是否允許異動修改浮點暫存器上下文，包括(例如)浮點暫存器及浮點控制暫存器；(c)是否允許異動修改存取暫存器(AR)；及(d)是否將封鎖某些程式例外條件免於造成中斷。若非受限異動中止，則可提供各種診斷資訊。舉例而言，起始非受限異動之最外部TBEGIN指令可指明程式指定異動診斷區塊(TDB)。另外，若異動分別歸因於程式中斷或造成解譯性執行結束之條件而中止，則亦可使用CPU之前置區域中或由主機之狀態描述指明的TDB。

上文指示各種暫存器型別。本文進一步詳細地解釋此等暫存器。一般暫存器可用作一般算術及邏輯運算中之累加器。在一實施例中，每一暫存器含有64個位元位置，且存在16個一般暫存器。該等一般暫存器係由數字0至15識別，且由一指令中之四位元R欄位指明。一些指令規定藉由具有若干R欄位來定址多個一般暫存器。對於一些指令，特定一般暫存器之使用被暗示，而非由該指令之R欄位明確地指明。

除了作為一般算術及邏輯運算中之累加器的使用以外，16個一般暫存器中之15個暫存器亦在位址產生中用作基底位址及索引暫存器。在此等狀況下，該等暫存器係由一指令中之四位元B欄位或X欄位指明。該B或X欄位中之值0指定出將不應用基底或索引，且因此，一般暫存器0將不被指明為含有基底位址或索引。

浮點指令使用浮點暫存器集合。在一實施例中，CPU具有16個浮

點暫存器。該等浮點暫存器係由數字0至15識別，且由浮點指令中之四位元R欄位指明。每一浮點暫存器為64個位元長，且可含有短(32位元)浮點運算元抑或長(64位元)浮點運算元。

浮點控制(FPC)暫存器為含有遮罩位元、旗標位元、資料例外碼及捨位模式位元之32位元暫存器，且在浮點運算之處理期間被使用。

另外，在一實施例中，CPU具有16個控制暫存器，每一控制暫存器具有64個位元位置。該等暫存器中之位元位置指派至系統中之特定設施，諸如，程式事件記錄(PER)(下文所論述)，且用以指定出可進行操作，抑或用以供給由該設施需要之特殊資訊。在一實施例中，對於異動設施，使用CR0(位元8及9)及CR2(位元61至63)，如下文所描述。

舉例而言，CPU具有被編號為0至15之16個存取暫存器。一存取暫存器由含有位址空間控制元素(ASCE)之間接規格的32個位元位置組成。位址空間控制元素為由動態位址轉譯(DAT)機制使用以將參考轉譯至對應位址空間之參數。當CPU處於被稱作存取暫存器模式之模式(由程式狀態字組(PSW)中之位元控制)時，用以指定針對儲存運算元參考之邏輯位址的指令B欄位指明存取暫存器，且由存取暫存器指定之位址空間控制元素係由DAT用於正被進行之參考。對於一些指令，使用R欄位而非B欄位。提供指令以用於載入及儲存存取暫存器之內容且用於將一個存取暫存器之內容移動至另一存取暫存器。

存取暫存器1至15中每一者可指明任何位址空間。存取暫存器0指明主要指令空間。當存取暫存器1至15中之一者用以指明位址空間時，CPU判定藉由轉譯存取暫存器之內容而指明哪一位址空間。當存取暫存器0用以指明位址空間時，CPU將該存取暫存器視為指明主要指令空間，且CPU並不檢驗該存取暫存器之實際內容。因此，16個存取暫存器可在任一時間指明主要指令空間及15個其他空間之最大值。

在一實施例中，存在多個位址空間型別。一位址空間為整數(虛擬位址)之連續序列，連同允許每一數字與儲存體中之位元組位置相關聯的特定變換參數。該序列以0開始且自左側至右側進行。

舉例而言，在z/Architecture中，當虛擬位址係由CPU使用以存取主儲存體(又名，主記憶體)時，虛擬位址係首先憑藉動態位址轉譯(DAT)而轉換至真實位址，且接著憑藉加前置詞(prefixing)而轉換至絕對位址。DAT可使用一個至五個表層級(頁、區段、區第三、區第二及區第一)作為變換參數。用於特定位址空間之最高層級表的指明(原點及長度)被稱作位址空間控制元素，且該指明被發現供DAT用於控制暫存器中或如由存取暫存器所指定。或者，用於位址空間之位址空間控制元素可為真實空間指明，其指示出DAT用以簡單地藉由將虛擬位址視為真實位址且不使用任何表來轉譯虛擬位址。

DAT在不同時間使用不同控制暫存器中或由存取暫存器指定之位址空間控制元素。選擇係由當前PSW中指定之轉譯模式判定。四個轉譯模式係可用的：主要空間模式、次要空間模式、存取暫存器模式，及本籍空間模式。取決於轉譯模式，不同位址空間係可定址的。

在CPU處於主要空間模式或次要空間模式時之任何時刻，CPU可轉譯屬於兩個位址空間(主要位址空間，及第二位址空間)之虛擬位址。在CPU處於存取暫存器模式時之任何時刻，CPU可轉譯高達16個位址空間(主要位址空間，及高達15個AR指定位址空間)之虛擬位址。在CPU處於本籍空間模式時之任何時刻，CPU可轉譯本籍位址空間之虛擬位址。

主要位址空間因而被識別，此係因為其由主要虛擬位址組成，該等主要虛擬位址係憑藉主要位址空間控制元素(ASCE)而轉譯。相似地，次要位址空間由憑藉次要ASCE轉譯之次要虛擬位址組成；AR指定位址空間由憑藉AR指定ASCE轉譯之AR指定虛擬位址組成；且

本籍位址空間由憑藉本籍ASCE轉譯之本籍虛擬位址組成。主要ASCE及次要ASCE分別處於控制暫存器1及7中。AR指定ASCE處於使用控制暫存器2、5及8經由被稱作存取暫存器轉譯(ART)之程序而定位的ASN第二表項中。本籍ASCE處於控制暫存器13中。

參看圖1來描述用以併入及使用本文所描述之異動設施之一或多項態樣的計算環境之一實施例。

參看圖1，在一實施例中，計算環境100係基於由紐約州阿蒙克市之 International Business Machines (IBM[®]) Corporation 提供的 z/Architecture。2010年8月第九版之名為「z/Architecture - Principles of Operation」的IBM公開案(公開案第SA22-7932-08號)中描述z/Architecture，該公開案之全文係據此以引用方式併入本文中。

Z/ARCHITECTURE、IBM以及Z/OS及Z/VM(下文所參考)為紐約州阿蒙克市之International Business Machines Corporation的註冊商標。本文所使用之其他名稱可為International Business Machines Corporation或其他公司之註冊商標、商標或產品名稱。

作為一實施例，計算環境100包括經由一或多個控制單元108耦接至一或多個輸入/輸出(I/O)器件106之中央處理器複合體(central processor complex, CPC)102。舉例而言，中央處理器複合體102包括一或多個中央處理器110、一或多個分割區112(例如，邏輯分割區(LP))、邏輯分割區超管理器114，及輸入/輸出子系統115，其中每一者在下文予以描述。

中央處理器110為分配給邏輯分割區之實體處理器資源。詳言之，每一邏輯分割區112具有一或多個邏輯處理器，其中每一邏輯處理器表示分配給該分割區之實體處理器110的全部或份額。特定分割區112之邏輯處理器可專用於該分割區，使得底層處理器資源110係針對彼分割區而保留；抑或該等邏輯處理器可與另一分割區共用，使得

該底層處理器資源潛在地可用於另一分割區。

邏輯分割區充當分離系統且具有一或多個應用程式，且視情況在其中具有駐留作業系統，該駐留作業系統針對每一邏輯分割區可不同。在一實施例中，作業系統為由紐約州阿蒙克市之International Business Machines Corporation提供的z/OS作業系統、z/VM作業系統、z/Linux作業系統或TPF作業系統。邏輯分割區112係由邏輯分割區超管理器114管理，邏輯分割區超管理器114係由處理器110上執行之韌體實施。如本文所使用，韌體包括(例如)處理器之微碼(microcode)及/或毫碼(millicode)。舉例而言，韌體包括在較高層級機器碼之實施中使用的硬體層級指令及/或資料結構。在一實施例中，舉例而言，韌體包括通常作為包括受信任軟體之微碼或特定於底層硬體之微碼進行遞送且控制對系統硬體之作業系統存取的專屬碼。

邏輯分割區及邏輯分割區超管理器各自包含駐留於與中央處理器相關聯之中央儲存體之各別分割區中的一或多個程式。邏輯分割區超管理器114之一實例為由紐約州阿蒙克市之International Business Machines Corporation提供的Processor Resource/System Manager(PR/SM)。

輸入/輸出子系統115在輸入/輸出器件106與主儲存體(又名，主記憶體)之間引導資訊流動。輸入/輸出子系統耦接至中央處理複合體，此在於：輸入/輸出子系統可為中央處理器複合體之部分或與中央處理器複合體分離。I/O子系統使中央處理器減輕直接地與輸入/輸出器件通信之任務的負擔，且准許資料處理與輸入/輸出處理並行地進行。為了提供通信，I/O子系統使用I/O通信配接器。存在各種通信配接器型別，包括(例如)通道、I/O配接器、PCI卡、乙太網路卡、小型電腦儲存介面(SCSI)卡等等。在本文所描述之特定實例中，I/O通信配接器為通道，且因此，I/O子系統在本文中被稱作通道子系統。然

而，此情形僅為一實例。可使用其他I/O子系統型別。

I/O子系統使用一或多個輸入/輸出路徑作為通信鏈路來管理至或自輸入/輸出器件106之資訊流動。在此特定實例中，此等路徑被稱作通道路徑，此係因為通信配接器為通道。

上文所描述之計算環境僅為可使用之計算環境之一實例。可使用包括但不限於非分割環境、其他分割環境及/或模擬環境之其他環境；實施例並不限於任一環境。

根據一或多項態樣，異動執行設施為CPU增強，其提供可用來使CPU執行指令序列(被稱作異動)之方式，該等指令可存取多個儲存位置，包括彼等位置之更新。如由其他CPU及I/O子系統所觀察，異動係(a)整體上作為單一不可部分完成操作而完成，抑或(b)中止，從而潛在地不留下異動曾經執行之跡象(惟本文所描述之某些條件除外)。因此，成功完成之異動可在沒有在傳統多處理模型中需要之任何特殊鎖定的情況下更新眾多儲存位置。

舉例而言，異動執行設施包括一或多個控制、一或多個指令、包括受限執行及非受限執行之異動處理，及中止處理，其中每一者在下文予以進一步描述。

在一實施例中，使用以下各者以控制異動執行設施：三個特殊用途控制，包括異動中止程式狀態字組(PSW)、異動診斷區塊(TDB)位址及異動巢狀結構深度；五個控制暫存器位元；及六個一般指令，包括TRANSACTION BEGIN(受限及非受限)、TRANSACTION END、EXTRACT TRANSACTION NESTING DEPTH、TRANSACTION ABORT及NONTRANSACTIONAL STORE。當該設施被安裝時，該設施安裝於(例如)呈該組態之所有CPU中。一設施指示(在一實施中為位元73)在為1時指示出異動執行設施被安裝。

當異動執行設施被安裝時，該組態提供非受限異動執行設施且

視情況提供受限異動執行設施，其中每一設施在下文予以描述。作為實例，當設施指示50及73皆為1時，受限異動執行設施被安裝。兩個設施指示皆在指定位置處儲存於記憶體中。

如本文所使用，指令名稱TRANSACTION BEGIN指代具有助憶鍵TBEGIN(用於非受限異動之異動開始)及TBEGINC(用於受限異動之異動開始)之指令。關於特定指令之論述係由指令名稱繼之以圓括號或括號中之助憶鍵指示，或簡單地由助憶鍵指示。

圖2A至圖2B中描繪TRANSACTION BEGIN(TBEGIN)指令之格式之一實施例。作為一實例，TBEGIN指令200包括：作業碼欄位202，其包括指定異動開始非受限操作之作業碼；基本欄位(B_1)204；位移欄位(D_1)206；及立即欄位(I_2)208。當 B_1 欄位為非零時，將由 B_1 204指定之一般暫存器的內容添加至 D_1 206以獲得第一運算元位址。

當 B_1 欄位為非零時，以下內容適用：

- 當異動巢狀結構深度最初為0時，第一運算元位址指明被稱作TBEGIN指定之TDB(下文進一步所描述)之256個位元組異動診斷區塊之位置，若異動中止，則各種診斷資訊可儲存至TBEGIN指定之TDB中。當CPU處於主要空間模式或存取暫存器模式時，第一運算元位址指明主要位址空間中之位置。當CPU處於次要空間模式或本籍空間模式時，第一運算元位址分別指明次要位址空間或本籍位址空間中之位置。當DAT關斷時，異動診斷區塊(TDB)位址(TDBA)指明真實儲存體中之位置。

判定對第一運算元之儲存可存取性。若為可存取，則運算元之邏輯位址置放至異動診斷區塊位址(TDBA)中，且TDBA有效。

- 當CPU已經處於非受限異動執行模式時，TDBA未被修改，且不可預測的是第一運算元是否針對可存取性而被測試。

當 B_1 欄位為0時，針對第一運算元未偵測到存取例外，且對於最

外部TBEGIN指令，TDBA無效。

在一實例中，I₂欄位之位元被定義如下：

一般暫存器儲存遮罩(GRSM)210(圖2B)：I₂欄位之位元0至7含有一般暫存器儲存遮罩(GRSM)。GRSM之每一位元表示一般暫存器之偶奇對，其中位元0表示暫存器0及1，位元1表示暫存器2及3，等等。當最外部TBEGIN指令之GRSM中的位元為0時，不儲存對應暫存器對。當最外部TBEGIN指令之GRSM中的位元為1時，將對應暫存器對儲存於不能由程式直接地存取之模型相依位置中。

若異動中止，則當執行最外部TBEGIN指令時，將已儲存暫存器對恢復為其內容。當異動中止時，不恢復所有其他(未儲存)一般暫存器之內容。

在惟最外部TBEGIN除外之所有TBEGIN上忽略一般暫存器儲存遮罩。

允許AR修改(A)212：A控制(I₂欄位之位元12)控制是否允許異動修改存取暫存器。有效允許AR修改控制為針對當前巢狀結構層級及針對所有外部層級的TBEGIN指令中之A控制之邏輯AND。

若有效A控制為0，則在嘗試修改任何存取暫存器時異動將用中止碼11(受限定指令)進行中止。若有效A控制為1，則在存取暫存器被修改(不存在任何其他中止條件)時異動將不中止。

允許浮點運算(F)214：F控制(I₂欄位之位元13)控制是否允許異動執行指定浮點指令。有效允許浮點運算控制為針對當前巢狀結構層級及針對所有外部層級的TBEGIN指令中之F控制之邏輯AND。

若有效F控制為0，則(a)在嘗試執行浮點指令時異動將用中止碼11(受限定指令)進行中止，且(b)浮點控制暫存器(FPCR)之位元組2中的資料例外碼(DXC)將不由任何資料例外程式例外條件設定。若有效F控制為1，則(a)在嘗試執行浮點指令(不存在任何其他中止條件)時異

動將不中止，且(b)FPCR中之DXC可由資料例外程式例外條件設定。

程式中斷篩選控制(PIFC)216： I_2 欄位之位元14至15為程式中斷篩選控制(PIFC)。PIFC控制在CPU處於異動執行模式時發生的某些類別之程式例外條件(例如，定址例外、資料例外、操作例外、保護例外等等)是否引起中斷。

有效PIFC為針對當前巢狀結構層級及針對所有外部層級的TBEGIN指令中之PIFC之最高值。當有效PIFC為0時，所有程式例外條件引起中斷。當有效PIFC為1時，具有異動執行類別1及2之程式例外條件引起中斷。(取決於例外之嚴重性，每一程式例外條件被指派至少一異動執行類別。嚴重性係基於在異動執行之重複執行期間復原的可能性，及作業系統是否需要查看中斷。)當有效PIFC為2時，具有異動執行類別1之程式例外條件引起中斷。保留為3之PIFC。

I_2 欄位之位元8至11(指令之位元40至43)被保留且應含有0；否則，程式在未來可不相容地操作。

參看圖3A至圖3B來描述異動開始受限(TBEGINC)指令之格式之一實施例。在一實施例中，TBEGINC 300包括：作業碼欄位302，其包括指定異動開始受限操作之作業碼；基本欄位(B_1)304；位移欄位(D_1)306；及立即欄位(I_2)308。將由 B_1 304指定之一般暫存器的內容添加至 D_1 306以獲得第一運算元位址。然而，在用異動開始受限指令的情況下，第一運算元位址不用以存取儲存體。取而代之，指令之 B_1 欄位包括0；否則，辨識規格例外。

在一實施例中， I_2 欄位包括各種控制，該等控制之實施例係在圖3B中予以描繪。

在一實施例中， I_2 欄位之位元被定義如下：

一般暫存器儲存遮罩(GRSM)310： I_2 欄位之位元0至7含有一般暫存器儲存遮罩(GRSM)。GRSM之每一位元表示一般暫存器之偶奇

對，其中位元0表示暫存器0及1，位元1表示暫存器2及3，等等。當GRSM中之位元為0時，不儲存對應暫存器對。當GRSM中之位元為1時，將對應暫存器對儲存於不能由程式直接地存取之模型相依位置中。

若異動中止，則當執行最外部TRANSACTION BEGIN指令時，將已儲存暫存器對恢復為其內容。當受限異動中止時，不恢復所有其他(未儲存)一般暫存器之內容。

當TBEGINC用以在非受限異動執行模式下繼續執行時，忽略一般暫存器儲存遮罩。

允許AR修改(A)312：A控制(I₂欄位之位元12)控制是否允許異動修改存取暫存器。有效允許AR修改控制為針對當前巢狀結構層級及針對任何外部TBEGIN或TBEGINC指令的TBEGINC指令之A控制中之邏輯AND。

若有效A控制為0，則在嘗試修改任何存取暫存器時異動將用中止碼11(受限定指令)進行中止。若有效A控制為1，則在存取暫存器被修改(不存在任何其他中止條件)時異動將不中止。

I₂欄位之位元8至11及13至15(指令之位元40至43及45至47)被保留且應含有0。

異動開始指令之結束係由TRANSACTION END(TEND)指令指定，該TRANSACTION END(TEND)指令之格式係在圖4中予以描繪。作為一實例，TEND指令400包括作業碼欄位402，作業碼欄位402包括指定異動結束操作之作業碼。

關於異動執行設施而使用數個術語，且因此，僅出於方便起見，下文以按字母排序次序提供術語清單。在一實施例中，此等術語具有以下定義：

中止：當異動在引起為0之異動巢狀結構深度的TRANSACTION

END指令之前結束時，異動中止。在一實施例中，當異動中止時，發生以下情形：

- 由異動之任何及所有層級進行之異動儲存存取被捨棄(亦即，未被交付)。
- 由異動之任何及所有層級進行之非異動儲存存取被交付。
- 由最外部TRANSACTION BEGIN指令之一般暫存器儲存遮罩(GRSM)指明的暫存器在異動執行之前恢復為其內容(亦即，恢復為其在執行最外部TRANSACTION BEGIN指令時之內容)。未由最外部TRANSACTION BEGIN指令之一般暫存器儲存遮罩指明的一般暫存器未被恢復。
- 存取暫存器、浮點暫存器及浮點控制暫存器未被恢復。當異動中止時，保持在異動執行期間對此等暫存器進行之任何改變。

異動可歸因於多種原因而中止，該等原因包括受限定指令之已嘗試執行、受限定資源之已嘗試修改、異動衝突、超過各種CPU資源、任何解譯性執行攔截條件、任何中斷、TRANSACTION ABORT指令，及其他原因。異動中止碼提供異動為何可中止之特定原因。

參看圖5來描述TRANSACTION ABORT(TABORT)指令之格式之一實例。作為一實例，TABORT指令500包括：作業碼欄位502，其包括指定異動中止操作之作業碼；基本欄位(B₂)504；及位移欄位(D₂)506。當B₂欄位為非零時，將由B₂ 504指定之一般暫存器的內容添加至D₂ 506以獲得第二運算元位址；否則，第二運算元位址係僅由D₂欄位形成，且忽略B₂欄位。第二運算元位址不用以定址資料；取而代之，該位址形成在中止處理期間置放於異動診斷區塊中之異動中止碼。用於第二運算元位址之位址計算遵循位址算術規則：在24位元定址模式下，將位元0至29設定為0；在31位元定址模式下，將位元0至32設定為0。

交付：在最外部TRANSACTION END指令完成時，CPU交付由異動進行之儲存存取(亦即，最外部異動及任何巢狀層級)，使得該等儲存存取可為其他CPU及I/O子系統所見。如由其他CPU且由I/O子系統所觀察，當發生交付時，由異動之所有巢狀層級進行之所有提取及儲存存取看來像是作為單一並行操作而發生。

一般暫存器、存取暫存器、浮點暫存器及浮點控制暫存器之內容未藉由交付程序修改。當交付異動之儲存時，保持在異動執行期間對此等暫存器進行之任何改變。

衝突：由一個CPU進行之異動存取與(a)由另一CPU進行之異動存取或非異動存取衝突，抑或與(b)由I/O子系統進行之非異動存取衝突(若兩種存取係針對同一快取行內之任何位置，且該等存取中之一者或多者為儲存)。

衝突可藉由CPU對指令之理論式執行而偵測，即使衝突可未以概念序列被偵測亦如此。

受限異動：受限異動為在受限異動執行模式下執行且經受以下限制之異動：

- 一般指令之子集係可用的。
- 可執行有限數目個指令。
- 可存取有限數目個儲存運算元位置。
- 異動限於單一巢狀結構層級。

在不存在重複中斷或與其他CPU或I/O子系統之衝突時，受限異動最終完成，因此無需中止處理常式。下文詳細地描述受限異動。

當在CPU已經處於非受限異動執行模式時執行TRANSACTION BEGIN受限(TBEGINC)指令時，執行作為巢狀非受限異動而繼續。

受限異動執行模式：當異動巢狀結構深度為0且異動係由TBEGINC指令起始時，CPU進入受限異動執行模式。在CPU處於受限

異動執行模式時，異動巢狀結構深度為1。

巢狀異動：當在CPU處於非受限異動執行模式時發出TRANSACTION BEGIN指令時，異動為巢狀。

異動執行設施使用被稱作扁平化巢狀結構之模型。在扁平化巢狀結構模式下，由內部異動進行之儲存不能由其他CPU及I/O子系統觀察直至最外部異動交付其儲存為止。相似地，若異動中止，則所有巢狀異動中止，且捨棄所有巢狀異動之所有異動儲存。

圖6中描繪巢狀異動之一實例。如圖所示，第一TBEGIN 600啟動最外部異動601，TBEGIN 602啟動第一巢狀異動，且TBEGIN 604啟動第二巢狀異動。在此實例中，TBEGIN 604及TEND 606定義最內部異動608。當TEND 610執行時，針對最外部異動及所有內部異動交付(612)異動儲存。

非受限異動：非受限異動為在非受限異動執行模式下執行之異動。儘管非受限異動並不以如同受限異動之方式受到限制，但非受限異動仍可歸因於多種原因而中止。

非受限異動執行模式：當異動係由TBEGIN指令起始時，CPU進入非受限異動執行模式。在CPU處於非受限異動執行模式時，異動巢狀結構深度可自1至最大異動巢狀結構深度而變化。

非異動存取：非異動存取為由CPU在不處於異動執行模式時進行之儲存運算元存取(亦即，在異動外之傳統儲存存取)。另外，由I/O子系統進行之存取為非異動存取。另外，NONTRANSACTIONAL STORE指令可用以在CPU處於非受限異動執行模式時造成非異動儲存存取。

參看圖7來描述NONTRANSACTIONAL STORE指令之格式之一實例。作為一實例，NONTRANSACTIONAL STORE指令700包括：複數個作業碼欄位702a、702b，其指定指明非異動儲存操作之作業

碼；暫存器欄位(R_1)704，其指定暫存器，該暫存器之內容被稱作第一運算元；索引欄位(X_2)706；基本欄位(B_2)708；第一位移欄位(DL_2)710；及第二位移欄位(DH_2)712。將由 X_2 欄位及 B_2 欄位指明之一般暫存器之內容添加至 DH_2 欄位及 DL_2 欄位之內容的串連之內容以形成第二運算元位址。當 X_2 欄位或 B_2 欄位中任一者或兩者為0時，對應暫存器不參與該添加。

使64位元第一運算元在第二運算元位置處非異動地置於不變。

由 DH_2 欄位及 DL_2 欄位之串連形成的位移被視為20位元帶正負號二進位整數。

第二運算元用以在雙字組邊界上對準；否則，辨識規格例外且抑制操作。

外部/最外部異動：具有較低編號異動巢狀結構深度之異動為外部異動。具有為1之異動巢狀結構深度值的異動為最外部異動。

最外部TRANSACTION BEGIN指令為在異動巢狀結構深度最初為0時執行之指令。最外部TRANSACTION END指令為使異動巢狀結構深度自1轉變至0之指令。在此實施例中，受限異動為最外部異動。

程式中斷篩選：當異動歸因於某些程式例外條件而中止時，程式可視情況防止發生中斷。此技術被稱作程式中斷篩選。程式中斷篩選經受中斷之異動類別、來自TRANSACTION BEGIN指令之有效程式中斷篩選控制，及控制暫存器0中之異動執行程式中斷篩選覆寫。

異動：在CPU處於異動執行模式時，異動包括所進行之儲存運算元存取及所變更之選擇之一般暫存器。對於非受限異動，儲存運算元存取可包括異動存取及非異動存取兩者。對於受限異動，儲存運算元存取限於異動存取。如由其他CPU及I/O子系統所觀察，由CPU在處於異動執行模式時進行之所有儲存運算元存取看來像是作為單一並行操作而發生。若異動中止，則捨棄異動儲存存取，且使由最外部

TRANSACTION BEGIN指令之一般暫存器儲存遮罩指明的任何暫存器恢復為其在異動執行之前的內容。

異動存取：異動存取為在CPU處於異動執行模式時進行之儲存運算元存取，惟由NONTRANSACTIONAL STORE指令進行之存取除外。

異動執行模式：術語異動執行模式描述非受限異動執行模式及受限異動執行模式兩者之共同操作。因此，當描述該操作時，術語非受限及受限用以限定異動執行模式。

當異動巢狀結構深度為0時，CPU不處於異動執行模式(亦被稱作非異動執行模式)。

如由CPU所觀察，在異動執行模式下進行之提取及儲存並非不同於在不處於異動執行模式時進行之提取及儲存。

在z/Architecture之一實施例中，異動執行設施係在控制暫存器0之位元8至9、控制暫存器2之位元61至63、異動巢狀結構深度、異動診斷區塊位址及異動中止程式狀態字組(PSW)的控制下。

在初始CPU重設之後，將控制暫存器0之位元位置8至9、控制暫存器2之位元位置62至63及異動巢狀結構深度的內容設定為0。當異動執行控制(控制暫存器0之位元8)為0時，不能將CPU置於異動執行模式。

下文描述關於各種控制之另外細節。

如所指示，異動執行設施係由控制暫存器0中之兩個位元及控制暫存器2中之三個位元控制。舉例而言：

控制暫存器0位元：在一實施例中，位元指派如下：

異動執行控制(TXC)：控制暫存器0之位元8為異動執行控制。此位元提供可用來使控制程式(例如，作業系統)指示異動執行設施是否可由該程式使用的機制。位元8將為1以成功地進入異動執行模式。

當控制暫存器0之位元8為0時，EXTRACT TRANSACTION NESTING DEPTH、TRANSACTION BEGIN及TRANSACTION END指令之已嘗試執行引起特殊操作執行。

參看圖8來描述EXTRACT TRANSACTION NESTING DEPTH(ETND)指令之格式之一實施例。作為一實例，EXTRACT TRANSACTION NESTING DEPTH指令800包括：作業碼欄位802，其指定指示擷取異動巢狀結構深度操作之作業碼；及暫存器欄位R₁ 804，其指明一般暫存器。

將當前異動巢狀結構深度置於一般暫存器R₁之位元48至63中。該暫存器之位元0至31保持不變，且將該暫存器之位元32至47設定為0。

在一另外實施例中，亦將最大異動巢狀結構深度置於一般暫存器R₁中，諸如，位元16至31中。

在一實施例中，EXTRACT TRANSACTION NESTING DEPTH指令使非受限異動能夠檢測當前異動巢狀結構深度值且將其值儲存於暫存器中。當前巢狀結構深度值維持於內部CPU計數器中。在非受限異動模式期間，當在無介入TEND的情況下執行TBEGIN時，累加計數器。既在判定適當碼定序時又對於診斷目的，此計數器可為程式所關注。

在一實施例中，若異動執行控制(控制暫存器0之位元8)為0，則辨識特殊操作例外且抑制操作。

條件碼：該碼保持不變。

程式例外：

- 操作(未安裝異動執行設施)
- 特殊操作
- 異動限制(歸因於受限定指令)

備註：為0之異動巢狀結構深度指示出CPU不處於異動執行模

式。

異動執行程式中斷篩選覆寫(PIFO)：控制暫存器0之位元9為異動執行程式中斷篩選覆寫。此位元提供可用來使控制程式確保在CPU處於異動執行模式時發生之任何程式例外條件引起中斷而不管由TRANSACTION BEGIN指令指定或暗示之有效程式中斷篩選控制的機制。

控制暫存器2位元：在一實施例中，指派如下：

異動診斷範疇(TDS)：控制暫存器2之位元61控制該暫存器之位元62至63中之異動診斷控制(TDC)的適用性，如下：

TDS

值	含義
0	TDC適用而不管CPU處於問題狀態抑或監督狀態。
1	僅當CPU處於問題狀態時TDC才適用。當CPU處於監督狀態時，處理如同TDC含有0一樣。

異動診斷控制(TDC)：控制暫存器2之位元62至63為可用以使異動出於診斷目的而隨機地中止之2位元無正負號整數。在一實施例中，TDC之編碼如下：

TDC

值	含義
0	正常操作；異動並不由於TDC而中止。
1	在隨機指令處但在最外部TRANSACTION END指令之執行之前中止每一異動。
2	在隨機指令處中止隨機異動。
3	保留

當異動歸因於非零TDC而中止時，則可發生以下情形中任一者：

- 將中止碼設定為碼7至11、13至16或255中任一者，其中該碼

之值係由CPU隨機地選擇；將條件碼設定成對應於中止碼。下文進一步描述中止碼。

- 對於非受限異動，將條件碼設定為1。在此狀況下，中止碼不適用。

無論是否實施TDC值1而皆模型相依。若未實施，則為1之值如同指定2一樣起作用。

對於受限異動，為1之TDC值被視為如同指定為2之TDC值一樣。

若指定為3之TDC值，則結果係不可預測的。

異動診斷區塊位址(TDBA)

當最外部TRANSACTION BEGIN(TBEGIN)指令之B₁欄位為非零時，自該指令之第一運算元位址設定有效異動診斷區塊位址(TDBA)。當CPU處於主要空間模式或存取暫存器模式時，TDBA指明主要位址空間中之位置。當CPU處於次要空間模式或本籍空間模式時，TDBA分別指明次要位址空間或本籍位址空間中之位置。當DAT(動態位址轉譯)關斷時，TDBA指明真實儲存體中之位置。

若異動隨後中止，則TDBA係由CPU使用以定位異動診斷區塊(被稱作TBEGIN指定之TDB)。TDBA之最右側三個位元為0，此意謂TBEGIN指定之TDB處於雙字組邊界上。

當最外部TRANSACTION BEGIN(TBEGIN)指令之B₁欄位為0時，異動診斷區塊位址無效，且若異動隨後中止，則不儲存TBEGIN指定之TDB。

異動中止PSW(TAPSW)

在巢狀結構深度最初為0時TRANSACTION BEGIN(TBEGIN)指令之執行期間，將異動中止PSW設定為當前PSW之內容；且異動中止PSW之指令位址指明下一循序指令(亦即，在最外部TBEGIN之後的指令)。在巢狀結構深度最初為0時TRANSACTION BEGIN受限

(TBEGINC)指令之執行期間，將異動中止PSW設定為當前PSW之內容，惟異動中止PSW之指令位址指明TBEGINC指令(而非在TBEGINC之後的下一循序指令)除外。

當異動中止時，用指示中止條件之嚴重性的碼來取代異動中止PSW中之條件碼。隨後，若異動歸因於不引起中斷之原因而中止，則自異動中止PSW載入PSW；若異動歸因於引起中斷之原因而中止，則將異動中止PSW儲存為中斷舊PSW。

在任何內部TRANSACTION BEGIN指令之執行期間不變更異動中止PSW。

異動巢狀結構深度(TND)

舉例而言，異動巢狀結構深度為每當用條件碼0來完成TRANSACTION BEGIN指令時累加且每當完成TRANSACTION END指令時遞減之16位元無正負號值。當異動中止時或由CPU重設時，將異動巢狀結構深度重設至0。

在一實施例中，實施為為15之最大TND。

在一實施中，當CPU處於受限異動執行模式時，異動巢狀結構深度為1。另外，儘管最大TND可被表示為4位元值，但TND經定義為16位元值以促進其在異動診斷區塊中之檢測。

異動診斷區塊(TDB)

當異動中止時，各種狀態資訊可儲存於異動診斷區塊(TDB)中，如下：

1. **TBEGIN指定之TDB**：對於非受限異動，當最外部TBEGIN指令之 B_1 欄位為非零時，該指令之第一運算元位址指明TBEGIN指定之TDB。此位址為可由應用程式之中止處理常式檢驗的應用程式指定位置。
2. **程式中斷(PI)TDB**：若非受限異動歸因於非篩選程式例外條

件而中止時，或若受限異動歸因於任何程式例外條件(亦即，引起正被辨識之程式中斷的任何條件)而中止時，將PI-TDB儲存至前置區域中之位置中。此情形可用於使作業系統在其可提供之任何診斷報告中進行檢測及登出。

3. **攔截TDB**：若異動歸因於引起攔截之任何程式例外條件而中止(亦即，該條件造成解譯性執行結束且造成控制傳回至主機程式)，則TDB儲存至用於客體作業系統之狀態描述區塊中指定的位置中。

在一實施例中，當TDB位址有效時(亦即，當最外部TBEGIN指令之B₁欄位為非零時)，僅儲存TBEGIN指定之TDB。

對於歸因於未篩選程式例外條件之中止，將儲存PI-TDB抑或攔截TDB中僅一者。因此，可存在針對一中止而儲存之零個、一個或兩個TDB。

下文描述關於TDB中每一者之一實例的另外細節：

TBEGIN指定之TDB：由有效異動診斷區塊位址指定之256位元組位置。當異動診斷區塊位址有效時，異動中止時儲存TBEGIN指定之TDB。TBEGIN指定之TDB經受在執行最外部TRANSACTION BEGIN指令時有效之所有儲存保護機制。用於TBEGIN指定之TDB之任何部分的PER(程式事件記錄)儲存變更事件係在執行最外部TBEGIN期間而非在異動中止處理期間被偵測。

PER之一目的係協助對程式進行除錯。作為實例，PER准許向程式提醒以下事件型別：

- 執行成功分支指令。提供使事件僅在分支目標位置處於已指明儲存區域內時才發生之選項。
- 自己指明儲存區域提取指令。
- 變更已指明儲存區域之內容。提供使事件僅在儲存區域處於已指明位址空間內時才發生之選項。

- 執行STORE USING REAL ADDRESS指令。
- 執行TRANSACTION END指令。

程式可選擇地指定出以上事件型別中之一或多者被辨識，惟用於STORE USING REAL ADDRESS之事件可僅與儲存變更事件一起被指定除外。關於PER事件之資訊係憑藉程式中斷而提供至程式，其中該中斷之原因係在中斷碼中被識別。

當異動診斷區塊位址無效時，不儲存TBEGIN指定之TDB。

程式中斷**TDB**：真實位置6,144至6,399(1800至18FF十六進位)。當異動歸因於程式中斷而中止時，儲存程式中斷TDB。當異動歸因於其他原因而中止時，程式中斷TDB之內容係不可預測的。

程式中斷TDB不經受任何保護機制。當在程式中斷期間儲存程式中斷TDB時，PER儲存變更事件不針對程式中斷TDB被偵測。

攔截TDB：由狀態描述之位置488至495指定的256位元組主機真實位置。當已中止異動引起客體程式中斷攔截(亦即，攔截碼8)時，儲存攔截TDB。當異動歸因於其他原因而中止時，攔截TDB之內容係不可預測的。攔截TDB不經受任何保護機制。

如圖9所描繪，在一實施例中，異動診斷區塊900之欄位如下：

格式902：位元組0含有有效性及格式指示，如下：

值	含義
0	TDB之剩餘欄位係不可預測的。
1	格式1 TDB，其剩餘欄位在下文予以描述。
2至255	保留

格式欄位為零之TDB被稱作空值TDB。

旗標904：位元組1含有各種指示，如下：

衝突符記有效性(CTV)：當異動歸因於提取或儲存衝突(亦即，分別為中止碼9或10)而中止時，位元組1之位元0為衝突符記有效性指

示。當CTV指示為1時，TDB之位元組16至23中的衝突符記910含有衝突被偵測之邏輯位址。當CTV指示為0時，TDB之位元組16至23係不可預測的。

當異動歸因於不同於擷取或儲存衝突之任何原因而中止時，將位元組1之位元0儲存為0。

受限異動指示(CTI)：當CPU處於受限異動執行模式時，將位元組1之位元1設定為1。當CPU處於非受限異動執行模式時，將位元組1之位元1設定為0。

保留：保留位元組1之位元2至7且將其儲存為0。

異動巢狀結構深度(TND)906：當異動中止時，位元組6至7含有異動巢狀結構深度。

異動中止碼(TAC)908：位元組8至15含有64位元無正負號異動中止碼。每一碼點指示異動被中止之原因。

當異動歸因於不同於程式中斷之條件而中止時，無論異動中止碼是否儲存於程式中斷TDB而皆模型相依。

衝突符記910：對於歸因於提取或儲存衝突(亦即，分別為中止碼9及10)而中止之異動，位元組16至23含有衝突被偵測之儲存位置的邏輯位址。當CTV位元(位元組1之位元0)為1時，衝突符記有意義。

當CTV位元為0時，位元組16至23係不可預測的。

由於由CPU進行之理論式執行，衝突符記可指明將未必由異動之概念執行序列存取的儲存位置。

已中止異動指令位址(ATIA)912：位元組24至31含有識別在偵測到中止時正執行之指令的指令位址。當異動歸因於中止碼2、5、6、11、13或256或更高之中止碼而中止時，或當異動歸因於中止碼4或13而中止且程式例外條件正設為空值時，ATIA直接地指向正被執行之指令。當異動歸因於中止碼4或12而中止且程式例外條件未設為空值

時，ATIA越過正被執行之指令。

當異動歸因於中止碼7至10、14至16或255而中止時，ATIA未必指示造成中止之確切指令，但可指向異動內之較早或較遲指令。

取決於如上文所描述之中止碼，若異動歸因於為執行型別指令之目標的指令而中止，則ATIA識別執行型別指令，從而指向或越過該指令。ATIA並不指示執行型別指令之目標。

當異動中止時，ATIA經受定址模式。在24位元定址模式下，欄位之位元0至40含有0。在31位元定址模式下，欄位之位元0至32含有0。

當異動歸因於不同於程式中斷之條件而中止時，無論已中止異動指令位址是否儲存於程式中斷TDB中而皆模型相依。

當異動歸因於中止碼4或12而中止且程式例外條件未設為空值時，ATIA並不指向造成中止之指令。藉由自ATIA減去由中斷長度碼(ILC)指示之半字組的數目，在正抑制或終止之條件下或針對正完成之非PER事件可識別造成中止之指令。當異動歸因於PER事件而中止且不存在其他程式例外條件時，ATIA係不可預測的。

當異動診斷區塊位址有效時，可在TBEGIN指定之TDB之位元組36至39中的程式中斷識別(PIID)中檢驗ILC。當篩選不適用時，可在真實儲存體中之位置140至143處的PIID中檢驗ILC。

例外存取識別(EAID)914：對於歸因於某些已篩選程式例外條件而中止之異動，TBEGIN指定之TDB之位元組32含有例外存取識別。在z/Architecture之一實例中，當例外條件引起中斷時，EAID之格式及EAID被儲存之狀況相同於真實位置160中描述的EAID之格式及EAID被儲存之狀況，如上文以引用方式併入之操作原理中所描述。

對於出於其他原因(包括引起程式中斷之任何例外條件)而中止之異動，位元組32係不可預測的。在程式中斷TDB中，位元組32係不可

預測的。

此欄位僅儲存於由異動診斷區塊位址指明之TDB中；否則，保留該欄位。EAID僅針對存取清單控制或DAT保護、ASCE型別、頁轉譯、區第一轉譯、區第二轉譯、區第三轉譯及區段轉譯程式例外條件被儲存。

資料例外碼(DXC)916：對於歸因於已篩選資料例外程式例外條件而中止之異動，TBEGIN指定之TDB之位元組33含有資料例外碼。在z/Architecture之一實例中，當例外條件引起中斷時，DXC之格式及DXC被儲存之狀況相同於真實位置147中描述的DXC之格式及DXC被儲存之狀況，如上文以引用方式併入之操作原理中所描述。在一實例中，位置147包括DXC。

對於出於其他原因(包括引起程式中斷之任何例外條件)而中止之異動，位元組33係不可預測的。在程式中斷TDB中，位元組33係不可預測的。

此欄位僅儲存於由異動診斷區塊位址指明之TDB中；否則，保留該欄位。僅針對資料程式例外條件而儲存DXC。

程式中斷識別(PIID)918：對於歸因於已篩選程式例外條件而中止之異動，TBEGIN指定之TDB之位元組36至39含有程式中斷識別。在z/Architecture之一實例中，當條件引起中斷時，PIID之格式相同於真實位置140至143中描述的PIID之格式(如上文以引用方式併入之操作原理中所描述)，惟PIID之位元13至14中的指令長度碼各別於例外條件被偵測之指令除外。

對於出於其他原因(包括引起程式中斷之例外條件)而中止之異動，位元組36至39係不可預測的。在程式中斷TDB中，位元組36至39係不可預測的。

此欄位僅儲存於由異動診斷區塊位址指明之TDB中；否則，保留

該欄位。程式中斷識別僅針對程式例外條件被儲存。

轉譯例外識別(TEID)920：對於歸因於以下已篩選程式例外條件中任一者而中止之異動，TBEGIN指定之TDB之位元組40至47含有轉譯例外識別。

- 存取清單控制或DAT保護
- ASCE型別
- 頁轉譯
- 區第一轉譯
- 區第二轉譯
- 區第三轉譯
- 區段轉譯例外

在z/Architecture之一實例中，當條件引起中斷時，TEID之格式相同於真實位置168至175中描述的TEID之格式，如上文以引用方式併入之操作原理中所描述。

對於出於其他原因(包括引起程式中斷之例外條件)而中止之異動，位元組40至47係不可預測的。在程式中斷TDB中，位元組40至47係不可預測的。

此欄位僅儲存於由異動診斷區塊位址指明之TDB中；否則，保留該欄位。

中斷事件(breaking event)位址922：對於歸因於已篩選程式例外條件而中止之異動，TBEGIN指定之TDB之位元組48至55含有中斷事件位址。在z/Architecture之一實例中，當條件引起中斷時，中斷事件位址之格式相同於真實位置272至279中描述的中斷事件位址之格式，如上文以引用方式併入之操作原理中所描述。

對於出於其他原因(包括引起程式中斷之例外條件)而中止之異動，位元組48至55係不可預測的。在程式中斷TDB中，位元組48至55

係不可預測的。

此欄位僅儲存於由異動診斷區塊位址指明之TDB中；否則，保留該欄位。

下文描述關於中斷事件之另外細節。

在z/Architecture之一實施例中，當安裝PER-3設施時，PER-3設施向程式提供最後指令之位址以造成CPU之循序執行的中斷。中斷事件位址記錄可用作針對萬用分支偵測之除錯協助。舉例而言，此設施提供CPU中之64位元暫存器，其被稱作中斷事件位址暫存器。每當不同於TRANSACTION ABORT之指令造成循序指令執行之中斷(亦即，PSW中之指令位址被取代，而非被累加該指令之長度)時，將彼指令之位址置放於中斷事件位址暫存器中。只要發生程式中斷，無論是否指示PER，皆將中斷事件位址暫存器之當前內容置放於真實儲存位置272至279中。

若造成中斷事件之指令為執行型別指令(EXECUTE或EXECUTE RELATIVE LONG)之目標，則將用以提取執行型別指令之指令位址置放於中斷事件位址暫存器中。

在z/Architecture之一實施例中，只要以下指令中之一者造成分支，就認為發生中斷事件：BRANCH AND LINK (BAL, BALR)；BRANCH AND SAVE (BAS, BASR)；BRANCH AND SAVE AND SET MODE (BASSM)；BRANCH AND SET MODE (BSM)；BRANCH AND STACK (BAKR)；BRANCH ON CONDITION (BC, BCR)；BRANCH ON COUNT (BCT, BCTR, BCTG, BCTGR)；BRANCH ON INDEX HIGH (BXH, BXHG)；BRANCH ON INDEX LOW OR EQUAL (BXLE, BXLEG)；BRANCH RELATIVE ON CONDITION (BRC)；BRANCH RELATIVE ON CONDITION LONG (BRCL)；BRANCH RELATIVE ON COUNT (BRCT, BRCTG)；BRANCH RELATIVE ON INDEX HIGH

(BRXH, BRXHG) ; BRANCH RELATIVE ON INDEX LOW OR EQUAL
 (BRXLE, BRXLG) ; COMPARE AND BRANCH (CRB, CGRB) ;
 COMPARE AND BRANCH RELATIVE (CRJ, CGRJ) ; COMPARE
 IMMEDIATE AND BRANCH (CIB, CGIB) ; COMPARE IMMEDIATE
 AND BRANCH RELATIVE (CIJ, CGIJ) ; COMPARE LOGICAL AND
 BRANCH (CLRB, CLGRB) ; COMPARE LOGICAL AND BRANCH
 RELATIVE (CLRJ, CLGRJ) ; COMPARE LOGICAL IMMEDIATE AND
 BRANCH (CLIB, CLGIB) ; 及 COMPARE LOGICAL IMMEDIATE AND
 BRANCH RELATIVE (CLIJ, CLGIJ) 。

只要以下指令中之一者完成，就亦認為發生中斷事件：
 BRANCH AND SET AUTHORITY (BSA) ; BRANCH IN SUBSPACE
 GROUP (BSG) ; BRANCH RELATIVE AND SAVE (BRAS) ; BRANCH
 RELATIVE AND SAVE LONG (BRASL) ; LOAD PSW (LPSW) ; LOAD
 PSW EXTENDED (LPSWE) ; PROGRAM CALL (PC) ; PROGRAM
 RETURN (PR) ; PROGRAM TRANSFER (PT) ; PROGRAM
 TRANSFER WITH INSTANCE (PTI) ; RESUME PROGRAM (RP) ; 及
 TRAP (TRAP2, TRAP4) 。

由於異動中止(隱含地抑或由於 TRANSACTION ABORT 指令)而
 不認為發生中斷事件。

模型相依診斷資訊924：位元組112至127含有模型相依診斷資
 訊。

對於惟12(已篩選程式中斷)除外之所有中止碼，將模型相依診斷
 資訊儲存於所儲存之每一TDB中。

在一實施例中，模型相依診斷資訊包括以下各者：

- 位元組112至119含有被稱作異動執行分支指示(TXBI)的為64
 個位元之向量。該向量之前63個位元中每一者指示在CPU處於異動執

行模式時執行分支指令之結果，如下：

值	含義
0	指令在無分支之情況下完成。
1	指令在有分支之情況下完成。

位元0表示第一此類分支指令之結果，位元1表示第二此類指令之結果，等等。

若在CPU處於異動執行模式時執行63個以下分支指令，則將不對應於分支指令之最右側位元設定為0(包括位元63)。當執行63個以上分支指令時，將TXBI之位元63設定為1。

TXBI中之位元係由如上文所列出的能夠造成中斷事件之指令設定，惟以下各者除外：

- 任何受限定指令並不使位元設定於TXBI中。
- 對於(例如)z/Architecture之指令，當BRANCH ON CONDITION、BRANCH RELATIVE ON CONDITION或BRANCH RELATIVE ON CONDITION LONG指令之M₁欄位為0時，或當以下指令之R₂欄位為0時，無論該指令之執行是否使位元設定於TXBI而皆模型相依。

- BRANCH AND LINK (BALR)；BRANCH AND SAVE (BASR)；BRANCH AND SAVE AND SET MODE (BASSM)；BRANCH AND SET MODE (BSM)；BRANCH ON CONDITION (BCR)；及BRANCH ON COUNT (BCTR, BCTGR)

- 對於由主機存取例外造成之中止條件，將位元組127之位元位置0設定為1。對於所有其他中止條件，將位元組127之位元位置0設定為0。

- 對於由載入/儲存單元(LSU)偵測之中止條件，位元組127之最右側五個位元含有原因之指示。對於未由LSU偵測之中止條件，保

留位元組127。

一般暫存器930：位元組128至255含有在異動中止時的一般暫存器0至15之內容。該等暫存器係以升序被儲存，其以位元組128至135中之一般暫存器0開始，接著為位元組136至143中之一般暫存器1，等等。

保留：保留所有其他欄位。除非另有指示，否則保留欄位之內容係不可預測的。

如由其他CPU及I/O子系統所觀察，在異動中止期間儲存TDB為在任何非異動儲存之後發生的多重存取參考。

異動可歸因於在異動執行之立即組態之範疇外的原因而中止。舉例而言，由超管理器(諸如，LPAR或z/VM)辨識之瞬態事件可造成異動中止。

提供於異動診斷區塊中之資訊意欲用於診斷目的且實質上正確。然而，因為中止可能已由在立即組態之範疇外的事件造成，所以諸如中止碼或程式中斷識別之資訊可能不準確地反映該組態內之條件，且因此不應被用來判定程式動作。

除了儲存於TDB中之診斷資訊以外，當異動歸因於任何資料例外程式例外條件而中止且AFP暫存器控制(控制暫存器0之位元45)及有效允許浮點運算控制(F)兩者皆為1時，將資料例外碼(DXC)置放至浮點控制暫存器(FPCR)之位元組2中，而不管篩選是否應用於程式例外條件。當異動中止且AFP暫存器控制或有效允許浮點運算控制中之一者或兩者為0時，不將DXC置放至FPCR中。

在一實施例中，如本文所指示，當安裝異動執行設施時，提供以下一般指令。

- EXTRACT TRANSACTION NESTING DEPTH
- NONTRANSACTIONAL STORE

- TRANSACTION ABORT
- TRANSACTION BEGIN
- TRANSACTION END

當CPU處於異動執行模式時，某些指令之已嘗試執行受到限定且造成異動中止。

當在受限異動執行模式下發出時，受限定指令之已嘗試執行亦可引起異動限制程式中斷，或可如同異動不受到限制一樣引起執行進行。

在z/Architecture之一實例中，作為實例，受限定指令包括以下非特殊權限指令：COMPARE AND SWAP AND STORE；MODIFY RUNTIME INSTRUMENTATION CONTROLS；PERFORM LOCKED OPERATION；PREFETCH DATA (RELATIVE LONG)，此時M₁欄位中之碼為6或7；STORE CHARACTERS UNDER MASK HIGH，此時M₃欄位為0且R₁欄位中之碼為6或7；STORE FACILITY LIST EXTENDED；STORE RUNTIME INSTRUMENTATION CONTROLS；SUPERVISOR CALL；及TEST RUNTIME INSTRUMENTATION CONTROLS。

在以上清單中，COMPARE AND SWAP AND STORE及PERFORM LOCKED OPERATION為可藉由在TX模式下使用基本指令而更有效率地實施之複雜指令。針對PREFETCH DATA及PREFETCH DATA RELATIVE LONG之狀況受到限定，此係由於為6及7之碼釋放快取行，從而使潛在地在異動完成之前交付資料成為必要。SUPERVISOR CALL受到限定，此係由於其造成中斷(此情形造成異動中止)。

在下文所列出之條件下，以下指令受到限定：

- BRANCH AND LINK (BALR)、BRANCH AND SAVE (BASR) 及BRANCH AND SAVE AND SET MODE，此時指令之R₂欄位為非零

且分支追蹤被啓用。

- BRANCH AND SAVE AND SET MODE及BRANCH AND SET MODE，此時R₂欄位爲非零且模式追蹤被啓用；SET ADDRESSING MODE，此時模式追蹤被啓用。
- MONITOR CALL，此時監視事件條件被辨識。

以上清單包括可形成追蹤項之指令。若此等指令被允許異動地執行且形成追蹤項，且異動隨後中止，則將推進控制暫存器12中之追蹤表指標，但將捨棄至追蹤表之儲存。此情形將在追蹤表中留下不一致間隙。因此，指令在其將形成追蹤項之狀況下受到限定。

當CPU處於異動執行模式時，無論以下指令是否受到限定而皆模型相依：CIPHER MESSAGE；CIPHER MESSAGE WITH CFB；CIPHER MESSAGE WITH CHAINING；CIPHER MESSAGE WITH COUNTER；CIPHER MESSAGE WITH OFB；COMPRESSION CALL；COMPUTE INTERMEDIATE MESSAGE DIGEST；COMPUTE LAST MESSAGE DIGEST；COMPUTE MESSAGE AUTHENTICATION CODE；CONVERT UNICODE-16 TO UNICODE-32；CONVERT UNICODE-16 TO UNICODE-8；CONVERT UNICODE-32 TO UNICODE-16；CONVERT UNICODE-32 TO UNICODE-8；CONVERT UNICODE-8 TO UNICODE-16；CONVERT UNICODE-8 TO UNICODE-32；PERFORM CRYPTOGRAPHIC COMPUTATION；RUNTIME INSTRUMENTATION OFF；及 RUNTIME INSTRUMENTATION ON。

以上指令中每一者當前係由硬體共處理器實施，抑或已處於過去機器中，且因此被認爲受到限定。

當有效允許AR修改(A)控制爲0時，以下指令受到限定：COPY ACCESS；LOAD ACCESS MULTIPLE；LOAD ADDRESS

EXTENDED；及SET ACCESS。

以上指令中每一者造成存取暫存器之內容被修改。若TRANSACTION BEGIN指令中之A控制為0，則程式已明確地指示出將不允許存取暫存器修改。

當有效允許浮點運算(F)控制為0時，浮點指令受到限定。

在某些情況下，以下指令可受到限定：EXTRACT CPU TIME；EXTRACT PSW；STORE CLOCK；STORE CLOCK EXTENDED；及STORE CLOCK FAST。

以上指令中每一者經受解譯性執行狀態描述中之攔截控制。若超管理器已設定針對此等指令之攔截控制，則該等指令之執行可歸因於超管理器實施而被拖延；因此，若發生攔截，可認為該等指令受到限定。

當非受限異動由於受限定指令之已嘗試執行而中止時，將異動診斷區塊中之異動中止碼設定為11(受限定指令)，且將條件碼設定為3，惟如下情形除外：當非受限異動歸因於原本將會引起特殊權限操作例外之指令之已嘗試執行而中止時，無論將中止碼設定為11(受限定指令)抑或4(由特殊權限操作程式中斷之辨識引起的未篩選程式中斷)而皆係不可預測的。當非受限異動歸因於在M₁欄位中之碼為6或7時PREFETCH DATA (RELATIVE LONG)之已嘗試執行或在M₃欄位為0且R₁欄位中之碼為6或7時STORE CHARACTERS UNDER MASK HIGH之已嘗試執行而中止時，無論將中止碼設定為11(受限定指令)抑或16(快取其他)而皆係不可預測的。當非受限異動歸因於MONITOR CALL之已嘗試執行而中止且存在監視事件條件及規格例外條件兩者時，無論將中止碼設定為11或4抑或(在程式中斷被篩選時)12而皆係不可預測的。

額外指令可在受限異動中受到限定。儘管此等指令當前經定義

為在非受限異動中不受到限定，但該等指令可在某些情況下在未來處理器上之非受限異動中受到限定。

可在未來處理器上之異動執行模式下允許某些受限定指令。因此，程式不應依賴於歸因於受限定指令之已嘗試執行而中止的異動。TRANSACTION ABORT指令應被用以可靠地造成異動中止。

在非受限異動中，程式應提供替代性非異動碼路徑以適應歸因於受限定指令而中止之異動。

在操作中，當異動巢狀結構深度為0時，引起條件碼0之TRANSACTION BEGIN(TBEGIN)指令之執行使CPU進入非受限異動執行模式。當異動巢狀結構深度為0時，引起條件碼0之TRANSACTION BEGIN受限(TBEGINC)指令之執行使CPU進入受限異動執行模式。

除了另有明確提及以外，應用於非異動執行之所有規則亦應用於異動執行。下文為在CPU處於異動執行模式時進行處理之額外特性。

當CPU處於非受限異動執行模式時，引起條件碼0之TRANSACTION BEGIN指令之執行使CPU保持於非受限異動執行模式。

如由CPU所觀察，在異動執行模式下進行之提取及儲存並非不同於在不處於異動執行模式時進行之提取及儲存。如由其他CPU及I/O子系統所觀察，在CPU處於異動執行模式時進行之所有儲存運算元存取看來像是單一區塊並行存取。亦即，對半字組、字組、雙字組或四倍字組內之所有位元組的存取經指定以看來像是區塊並行，如由其他CPU及I/O(例如，通道)程式所觀察。半字組、字組、雙字組或四倍字組在此章節中被稱作區塊。當提取型別參考經指定以在區塊內看來像是並行時，在該區塊中含有之位元組正被提取的時間期間不准許由另

一CPU或I/O程式對該區塊之儲存存取。當儲存型別參考經指定以在區塊內看來像是並行時，在該區塊內之位元組正被儲存的時間期間不准許由另一CPU或I/O程式對該區塊之存取(提取抑或儲存)。

針對指令之儲存存取以及DAT及ART(存取暫存器表)表提取遵循非異動規則。

CPU憑藉使異動巢狀結構深度轉變至0之TRANSACTION END指令而正常地離開異動執行模式，在該狀況下，異動完成。

當CPU憑藉TRANSACTION END指令之完成而離開異動執行模式時，交付在處於異動執行模式時進行之所有儲存；亦即，該等儲存看來像是作為單一區塊並行操作而發生，如由其他CPU及I/O子系統所觀察。

異動可出於多種原因而隱含地中止，或異動可由TRANSACTION ABORT指令明確地中止。下文描述異動中止、對應中止碼及置放至異動中止PSW中之條件碼的實例可能原因。

外部中斷：將異動中止碼設定為2，且將異動中止PSW中之條件碼設定為2。作為外部中斷處理之部分，將異動中止PSW儲存為外部舊PSW。

程式中斷(未篩選)：引起中斷之程式例外條件(亦即，未篩選條件)使異動用碼4而中止。將異動中止PSW中之條件碼設定為特定於程式中斷碼。作為程式中斷處理之部分，將異動中止PSW儲存為程式舊PSW。

原本將會引起異動歸因於操作例外而中止之指令可得到替代結果：對於非受限異動，異動可代替地用中止碼11(受限定指令)而中止；對於受限異動，可辨識異動限制程式中斷，而非操作例外。

當PER(程式事件記錄)事件結合任何其他未篩選程式例外條件被辨識時，將條件碼設定為3。

機器檢查中斷：將異動中止碼設定為5，且將異動中止PSW中之條件碼設定為2。作為機器檢查中斷處理之部分，將異動中止PSW儲存為機器檢查舊PSW。

I/O中斷：將異動中止碼設定為6，且將異動中止PSW中之條件碼設定為2。作為I/O中斷處理之部分，將異動中止PSW儲存為I/O舊PSW。

提取溢位：當異動嘗試自多於CPU所支援之位置的位置進行提取時，偵測提取溢位條件。將異動中止碼設定為7，且將條件碼設定為2或3。

儲存溢位：當異動嘗試儲存至多於CPU所支援之位置的位置時，偵測儲存溢位條件。將異動中止碼設定為8，且將條件碼設定為2或3。

回應於提取或儲存溢位中止而允許條件碼為2或3會允許CPU指示潛在可重試情形(例如，條件碼2指示異動之重新執行可有生產力；而條件碼3並不建議重新執行)。

提取衝突：當另一CPU或I/O子系統嘗試儲存至已由此CPU異動地提取之位置中時，偵測提取衝突條件。將異動中止碼設定為9，且將條件碼設定為2。

儲存衝突：當另一CPU或I/O子系統嘗試存取在由此CPU進行之異動執行期間已被儲存之位置時，偵測儲存衝突條件。將異動中止碼設定為10，且將條件碼設定為2。

受限定指令：當CPU處於異動執行模式時，受限定指令之已嘗試執行造成異動中止。將異動中止碼設定為11，且將條件碼設定為3。

當CPU處於受限異動執行模式時，無論受限定指令之已嘗試執行引起異動限制程式中斷抑或歸因於受限定指令之中止而皆係不可預測的。異動仍中止，但中止碼可指示任一原因。

程式例外條件(已篩選)：不引起中斷之程式例外條件(亦即，已篩選之條件)造成異動用為12之異動中止碼而中止。將條件碼設定為3。

巢狀結構深度超過：當異動巢狀結構深度處於該組態之最大可允許值且執行TRANSACTION BEGIN指令時，偵測巢狀結構深度超過條件。使異動用為13之異動中止碼而中止，且將條件碼設定為3。

快取提取相關條件：由CPU之快取電路偵測與由異動提取之儲存位置相關的條件。使異動用為14之異動中止碼而中止，且將條件碼設定為2抑或3。

快取儲存相關條件：由CPU之快取電路偵測與由異動儲存之儲存位置相關的條件。使異動用為15之異動中止碼而中止，且將條件碼設定為2抑或3。

快取其他條件：由CPU之快取電路偵測快取其他條件。使異動用為16之異動中止碼而中止，且將條件碼設定為2抑或3。

在異動執行期間，若CPU使用映射至相同絕對位址之不同邏輯位址來存取指令或儲存運算元，則無論異動是否中止而皆模型相依。若異動歸因於使用映射至相同絕對位址之不同邏輯位址的存取而中止，則取決於條件，設定中止碼14、15或16。

雜項條件：雜項條件為造成異動中止的由CPU辨識之任何其他條件。將異動中止碼設定為255，且將條件碼設定為2抑或3。

當多個組態正在同一機器(例如，邏輯分割區或虛擬機)中執行時，異動可歸因於不同組態中發生之外部機器檢查或I/O中斷而中止。

儘管上文提供實例，但可提供具有對應中止碼及條件碼之異動中止的其他原因。舉例而言，一原因可為重新啟動中斷，其中將異動中止碼設定為1，且將異動中止PSW中之條件碼設定為2。作為重新啓

動處理之部分，將異動中止PSW儲存為重新啓動舊PSW。作為一另外實例，一原因可為監督呼叫條件，其中將中止碼設定為3，且將異動中止PSW中之條件碼設定為3。其他或不同實例亦係可能的。

備註：

1. 雜項條件可由以下各者中任一者引起：
 - 諸如在 z/Architecture 中之指令 COMPARE AND REPLACE DAT TABLE ENTRY、COMPARE AND SWAP AND PURGE、INVALIDATE DAT TABLE ENTRY、INVALIDATE PAGE TABLE ENTRY、NQ 控制為 0 且 SK 控制為 1 的 PERFORM FRAME MANAGEMENT FUNCTION、NQ 控制為 0 的 SET STORAGE KEY EXTENDED，其係由該組態中之另一 CPU 執行；將條件碼設定為 2。
 - 在 CPU 上執行諸如重設、重新啓動或停止之運算子函式，或等效 SIGNAL PROCESSOR 命令。
 - 上文未列舉之任何其他條件；將條件碼設定為 2 或 3。
2. 提取及儲存衝突被偵測之位置可為同一快取行內之任何地方。
3. 在某些條件下，CPU 可不能區分相似中止條件。舉例而言，提取或儲存溢位可不能與各別提取或儲存衝突區分開。
4. 由 CPU 對多個指令路徑之理論式執行可引起異動歸因於衝突或溢位條件而中止，即使此類條件並不以概念序列而發生亦如此。在處於受限異動執行模式時，CPU 可臨時地禁止理論式執行，從而允許異動嘗試在不理論式地偵測此類衝突或溢位的情況下完成。

TRANSACTION ABORT 指令之執行造成異動中止。自第二運算元位址設定異動中止碼。取決於第二運算元位址之位元 63 為 0 抑或 1，將條件碼分別設定為 2 抑或 3。

圖 10 概述儲存於異動診斷區塊中之實例中止碼，及對應條件碼

(CC)。圖10中之描述說明一特定實施。值之其他實施及編碼係可能的。

在一實施例中且如上文所提及，異動設施規定受限異動及非受限異動兩者，以及與其相關聯之處理。最初論述受限異動，且接著論述非受限異動。

受限異動在無後援路徑的情況下執行於異動模式下。其為用於緊湊函式之處理的模式。在不存在重複中斷或與其他CPU或I/O子系統之衝突(亦即，由將不允許異動成功地完成之條件造成)時，受限異動最終將完成；因此，無需且不指定中止處理常式。舉例而言，在不存在以下各者時，異動最終將完成：不能處理之條件(例如，除以0)的違反、不允許異動完成之條件(例如，不允許指令執行之計時器中斷、熱I/O等等)，或與受限異動相關聯之限定或限制的違反。

當異動巢狀結構深度最初為0時，受限異動係由TRANSACTION BEGIN受限(TBEGINC)指令起始。在一實施例中，受限異動經受以下限制。

1. 異動執行不多於32個指令，不包括TRANSACTION BEGIN受限(TBEGINC)及TRANSACTION END指令。

2. 異動中之所有指令將在儲存體之256個連續位元組內，包括TRANSACTION BEGIN受限(TBEGINC)及任何TRANSACTION END指令。

3. 除了受限定指令以外，以下限定亦應用於受限異動。

a. 指令限於被稱作一般指令之指令，包括(例如)加法、減法、乘法、除法、移位、旋轉等等。

b. 分支指令限於以下各者(在一實例中，所列之指令屬於z/Architecture)：

- BRANCH RELATIVE ON CONDITION，其中 M_1 為非零且 RI_2

欄位含有正值。

- BRANCH RELATIVE ON CONDITION LONG，其中M₁欄位為非零，且RI₂欄位含有不造成位址環繞之正值。

- COMPARE AND BRANCH RELATIVE、COMPARE IMMEDIATE AND BRANCH RELATIVE、COMPARE LOGICAL AND BRANCH RELATIVE，及COMPARE LOGICAL IMMEDIATE AND BRANCH RELATIVE，其中M₃欄位為非零且RI₄欄位含有正值。(亦即，僅具有非零分支遮罩之前向分支。)

c. 除了TRANSACTION END及造成已指定運算元序列化之指令以外，造成序列化功能之指令受到限定。

d. 儲存體間操作(SS-)及具有延伸作業碼之儲存體間操作(SSE-)指令受到限定。

e. 所有以下一般指令(在此實例中，其屬於z/Architecture)受到限定：CHECKSUM；CIPHER MESSAGE；CIPHER MESSAGE WITH CFB；CIPHER MESSAGE WITH CHAINING；CIPHER MESSAGE WITH COUNTER；CIPHER MESSAGE WITH OFB；COMPARE AND FORM CODEWORD；COMPARE LOGICAL LONG；COMPARE LOGICAL LONG EXTENDED；COMPARE LOGICAL LONG UNICODE；COMPARE LOGICAL STRING；COMPARE UNTIL SUBSTRING EQUAL；COMPRESSION CALL；COMPUTE INTERMEDIATE MESSAGE DIGEST；COMPUTE LAST MESSAGE DIGEST；COMPUTE MESSAGE AUTHENTICATION CODE；CONVERT TO BINARY；CONVERT TO DECIMAL；CONVERT UNICODE-16 TO UNICODE-32；CONVERT UNICODE-16 TO UNICODE-8；CONVERT UNICODE-32 TO UNICODE-16；CONVERT UNICODE-32 TO UNICODE-8；CONVERT UNICODE-8 TO

UNICODE-16 ; CONVERT UNICODE-8 TO UNICODE-32 ; DIVIDE ;
 DIVIDE LOGICAL ; DIVIDE SINGLE ; EXECUTE ; EXECUTE
 RELATIVE LONG ; EXTRACT CACHE ATTRIBUTE ; EXTRACT CPU
 TIME ; EXTRACT PSW ; EXTRACT TRANSACTION NESTING
 DEPTH ; LOAD AND ADD ; LOAD AND ADD LOGICAL ; LOAD
 AND AND ; LOAD AND EXCLUSIVE OR ; LOAD AND OR ; LOAD
 PAIR DISJOINT ; LOAD PAIR FROM QUADWORD ; MONITOR
 CALL ; MOVE LONG ; MOVE LONG EXTENDED ; MOVE LONG
 UNICODE ; MOVE STRING ; NON-TRANSACTIONAL STORE ;
 PERFORM CRYPTOGRAPHIC COMPUTATION ; PREFETCH DATA ;
 PREFETCH DATA RELATIVE LONG ; RUNTIME
 INSTRUMENTATION EMIT ; RUNTIME INSTRUMENTATION
 NEXT ; RUNTIME INSTRUMENTATION OFF ; RUNTIME
 INSTRUMENTATION ON ; SEARCH STRING ; SEARCH ; STRING
 UNICODE ; SET ADDRESSING MODE ; STORE CHARACTERS
 UNDER MASK HIGH，此時M₃欄位為0且R₁欄位中之碼為6或7；
 STORE CLOCK ; STORE CLOCK EXTENDED ; STORE CLOCK
 FAST ; STORE FACILITY LIST EXTENDED ; STORE PAIR TO
 QUADWORD ; TEST ADDRESSING MODE ; TRANSACTION
 ABORT ; TRANSACTION BEGIN(TBEGIN 及 TBEGINC 兩者)；
 TRANSLATE AND TEST EXTENDED ; TRANSLATE AND TEST
 REVERSE EXTENDED ; TRANSLATE EXTENDED ; TRANSLATE
 ONE TO ONE ; TRANSLATE ONE TO TWO TRANSLATE TWO TO
 ONE ; 及 TRANSLATE TWO TO TWO。

4. 異動之儲存運算元存取不多於四個八倍字組。備註：LOAD ON CONDITION及STORE ON CONDITION被看作參考儲存體而不管

條件碼。舉例而言，八倍字組為32位元組邊界上之32個連續位元組的群組。

5. 在此CPU上執行之異動或由其他CPU或I/O子系統進行之儲存並不存取任何4 K位元組區塊中之儲存運算元，該等4 K位元組區塊含有儲存體之256個位元組，其以TRANSACTION BEGIN受限(TBEGINC)指令開始。

6. 異動不使用映射至相同絕對位址之不同邏輯位址來存取指令或儲存運算元。

7. 由異動進行之運算元參考將在單一雙字組內，惟對於LOAD ACCESS MULTIPLE、LOAD MULTIPLE、LOAD MULTIPLE HIGH、STORE ACCESS MULTIPLE、STORE MULTIPLE及STORE MULTIPLE HIGH除外，運算元參考將在單一八倍字組內。

若受限異動違反上文所列出之限制1至7中任一者，則(a)辨識異動限制程式中斷，或(b)執行如同異動未受到限制一樣行進，惟另外限制違反仍可引起異動受限程式中斷除外。採取哪一動作係不可預測的，且所採取之動作可基於哪一限制被違反而不同。

在不存在限制違反、重複中斷或與其他CPU或I/O子系統之衝突時，受限異動最終將完成，如上文所描述。

1. 若異動滿足以下準則，則成功地完成受限異動之機會改良：

- a. 所發出之指令少於為32之最大值。
- b. 儲存運算元參考少於為4個八倍字組之最大值。
- c. 儲存運算元參考處於同一快取行上。
- d. 對相同位置之儲存運算元參考藉由所有異動以相同次序而發生。

2. 受限異動未必確信在其第一執行時成功地完成。然而，若未違反所列出限制中任一者之受限異動中止，則CPU使用確保異動之重

複執行隨後成功的電路。

3. 在受限異動內，TRANSACTION BEGIN為受限指令，因此，受限異動不能為巢狀。

4. 由受限異動違反以上限制1至7中任一者可引起程式迴圈。

5. 受限異動之限制相似於比較及交換迴圈之限制。由於來自其他CPU及I/O子系統之潛在干擾，不存在COMPARE AND SWAP指令將用條件碼0永遠完成之架構確信。受限異動可遭受呈提取或儲存衝突中止或熱中斷之形式的相似干擾。

CPU使用公平演算法以確保在不存在任何限制違反時受限異動最終完成。

6. 為了判定完成受限異動所需要之重複反覆的數目，程式可在不經受一般暫存器儲存遮罩之一般暫存器中使用計數器。下文展示一實例。

	LH1	15,0	零重試計數器。
迴圈	TBEGINC	0(0),X 'FE00'	保留GR 0至13
	AHI	15,1	累加計數器
	...		
	...	受限異動執行碼	
	...		
	TEND		異動之結束。

* R15現在含有重複異動嘗試之計數。

應注意，在此實例中，兩個暫存器14及15皆未被恢復。亦應注意，在一些模型上，若CPU在TBEGINC指令完成之後但在AHI指令完成之前偵測到中止條件，則一般暫存器15中之計數可低。

如由CPU所觀察，在異動執行模式下進行之提取及儲存並非不同於在不處於異動執行模式時進行之提取及儲存。

在一實施例中，使用者(亦即，建立異動之使用者)選擇異動是否將受到限制。參看圖11來描述與受限異動之處理相關聯之邏輯且尤其是與TBEGINC指令相關聯之處理的一實施例。TBEGINC指令之執行使CPU進入受限異動執行模式或保持於非受限執行模式。執行TBEGINC之CPU(亦即，處理器)執行圖11之邏輯。

參看圖11，基於TBEGINC指令之執行，執行序列化功能(步驟1100)。序列化功能或操作包括在發生概念上後續之儲存存取(及相關參考位元及改變位元設定)之前由CPU完成所有概念上先前之儲存存取(及作為一實例，對於z/Architecture，相關參考位元及改變位元設定)，如由其他CPU及I/O子系統所觀察。序列化影響對儲存體及對儲存體金鑰之所有CPU存取的序列，惟與ART表項及DAT表項提取相關聯之存取除外。

如由處於異動執行模式之CPU所觀察，序列化正常地操作(如上文所描述)。如由其他CPU及I/O子系統所觀察，由於使異動巢狀結構深度遞減至零(正常結束)之TRANSACTION END指令抑或由於正被中止之異動，在CPU處於異動執行模式時執行之序列化操作在CPU離開異動執行模式時發生。

在執行序列化之後，進行關於是否辨識例外之判定(查詢1102)。若是，則處置例外(步驟1104)。舉例而言，若異動執行控制(控制暫存器0之位元8)為0，則辨識特殊操作例外且抑制操作。作為另外實例，若B₁欄位(指令之位元16至19)為非零，則辨識規格例外且抑制操作；若TBEGINC為執行型別指令之目標，則辨識執行例外且抑制操作；且若於組態中未安裝異動執行設施，則辨識操作例外且抑制操作。若CPU已經處於受限異動執行模式，則辨識異動受限例外程式例外且抑制操作。另外，若異動巢狀結構深度在累加1時將超過模型相依最大異動巢狀結構深度，則用中止碼13而中止異動。可辨識及處置其他或

不同例外。

然而，若不存在例外，則進行關於異動巢狀結構深度是否為0之判定(查詢1106)。若異動巢狀結構深度為0，則將異動診斷區塊位址看作無效(步驟1108)；自當前PSW之內容設定異動中止PSW，惟異動中止PSW之指令位址指明TBEGINC指令而非下一循序指令除外(步驟1110)；且將如由一般暫存器儲存遮罩指明之一般暫存器對的內容儲存於不能由程式直接地存取之模型相依位置中(步驟1112)。另外，將巢狀結構深度設定為1(步驟1114)。另外，將允許浮點運算(F)及程式中斷篩選控制(PIFC)之有效值設定為0(步驟1316)。另外，判定允許AR修改(A)控制(指令之I₂欄位的位元12欄位)之有效值(步驟1118)。舉例而言，有效A控制為針對當前層級及針對任何外部TBEGIN指令之TBEGINC指令中之A控制的邏輯AND。

返回至查詢1106，若異動巢狀結構深度大於0，則使巢狀結構深度累加1(步驟1120)。另外，將允許浮點運算(F)之有效值設定為0，且程式中斷篩選控制(PIFC)之有效值不變(步驟1122)。處理接著繼續步驟1118。在一實施例中，異動之成功起始引起條件碼0。此情形推斷出與執行TBEGINC指令相關聯之邏輯之一實施例。

在一實施例中，上文所提供之例外檢查可以變化次序而發生。用於例外檢查之一特定次序如下：

優先權相同於針對一般狀況之程式中斷條件之優先權的例外。

歸因於含有非零值之B₁欄位的規格例外。

歸因於超過異動巢狀結構深度之中止。

歸因於正常完成之條件碼0。

另外，以下各者在一或多個實施例中適用：

1. 若異動並不在異動憑藉TRANSACTION END而正常地結束時中止，則僅恢復經指明為由一般暫存器儲存遮罩儲存之暫存器。在中

止時僅恢復由最外部TRANSACTION BEGIN指令之GRSM指明的暫存器。

I₂欄位應指明提供由受限異動改變之輸入值的所有暫存器對。因此，若異動中止，則當重新執行受限異動時，輸入暫存器值將恢復為其原始內容。

2. 在大多數模型上，既在TRANSACTION BEGIN上又在異動中止時，可藉由指定在一般暫存器儲存遮罩中進行儲存及恢復所需要之暫存器的最小數目而實現改良型效能。

3. 以下情形說明基於當前異動巢狀結構深度(TND)且在TND為非零時的TRANSACTION BEGIN指令(TBEGIN及TBEGINC兩者)之結果，而無論CPU處於非受限異動執行模式抑或受限異動執行模式：

指令 TND=0

TBEGIN 進入非受限異動執行模式

TBEGINC 進入受限異動執行模式

指令 TND>0

	NTX模式	CTX模式
TBEGIN	在非受限異動執行模式下繼續	異動受限例外
TBEGINC	在非受限異動執行模式下繼續	異動受限例外

解釋：

CTX CPU處於受限異動執行模式

NTX CPU處於非受限異動執行模式

TND 指令開始時之異動巢狀結構深度

如本文所描述，在一態樣中，假定受限異動不含有使其不能夠完成之條件，則受限異動確信完成。為了確保受限異動完成，執行該異動之處理器(例如，CPU)可採取某些動作。舉例而言，若受限異動具有中止條件，則CPU可臨時地：

- (a)禁止無序執行；
- (b)禁止其他CPU存取衝突儲存位置；
- (c)誘發中止處理之隨機延遲；及/或

(d)調用其他措施以促進成功完成。

概言之，受限異動之處理如下：

- 若已經處於受限TX模式，則辨識異動受限例外。
- 若當前TND(異動巢狀結構深度) > 0，則執行如同非受限異

動一樣行進

- 將有效F控制設定為0
- 有效PIFC不變
- 允許最外部非受限TX呼叫可或不使用受限TX之服務功

能。

- 若當前TND = 0，則：
 - 異動診斷區塊位址無效。
 - 中止時未儲存指令指定之TDB
 - 將異動中止PSW設定為TBEGINC之位址
 - 並非下一循序指令
 - 儲存於不能由程式存取之模型相依位置中的由GRSM指明之

一般暫存器對

◦ 視情況而形成(自D₂運算元)之異動符記。異動符記為異動之識別符。異動符記可等於儲存運算元位址或另一值。

- 有效A = TBEGINC A及任何外部A
- 累加TND
- 若TND自0轉變至1，則CPU進入受限TX模式
- 否則，CPU操持於非受限TX模式
- 指令用CC0而完成
- 例外：
 - 在B₁欄位為非零時之規格例外(PIC(程式中斷碼)0006)
 - 在異動執行控制(CR0.8)為0時之特殊操作例外(PIC 0013十六

進位)

- 在受限TX模式下發出時之異動受限例外(PIC 0018十六進位)
- 在未安裝受限異動執行設施時之操作例外(PIC 0001)
- 在指令為執行型別指令之目標時的執行例外(PIC 0003)
- 在巢狀結構深度超過時之中止碼13
- 受限異動中之中止條件：
- 中止PSW指向TBEGINC指令
- 並非在TBEGINC指令之後的指令
- 中止條件造成整個TX被重新驅動

*無失敗路徑

- CPU採取特殊措施以確保在重新驅動時之成功完成
- 假定無持久衝突、中斷或受限違反，則異動確信最終完成。
- 限制違反：
- PIC 0018十六進位一指示異動限制之違反
- 或，異動如同非受限一樣執行

如上文所描述，除了為選用之受限異動處理以外，在一實施例中，異動設施亦提供非受限異動處理。參看圖12來描述關於非受限異動之處理且尤其是與TBEGIN指令相關聯之處理的另外細節。TBEGIN指令之執行使CPU進入或保持於非受限異動執行模式。執行TBEGIN之CPU(亦即，處理器)執行圖12之邏輯。

參看圖12，基於TBEGIN指令之執行，執行序列化功能(上文所描述)(步驟1200)。在執行序列化之後，進行關於是否辨識例外之判定(查詢1202)。若是，則處置例外(步驟1204)。舉例而言，若異動執行控制(控制暫存器0之位元8)為0，則辨識特殊操作例外且抑制操作。另外，若程式中斷篩選控制(指令之I₂欄位的位元14至15)含有值3，則辨識規格例外且抑制操作；或第一運算元位址並不指明雙字組邊界。

若於組態中未安裝異動執行設施，則辨識操作例外且抑制操作；且若TBEGI為執行型別指令之目標，則辨識執行例外且抑制操作。另外，若CPU處於受限異動執行模式，則辨識異動受限例外程式例外且抑制操作。另外，若異動巢狀結構深度在累加1時將超過模型相依最大異動巢狀結構深度，則用中止碼13而中止異動。

再者，當指令之 B_1 欄位為非零且CPU並不處於異動執行模式(亦即，異動巢狀結構深度為0)時，則判定對第一運算元之儲存可存取性。取決於特定存取例外條件，若第一運算元不能被存取以進行儲存，則辨識存取例外且操作被設為空值、抑制抑或終止。另外，辨識用於第一運算元之任何PER儲存變更事件。當 B_1 欄位為非零且CPU已經處於異動執行模式時，是否判定對第一運算元之儲存可存取性係不可預測的，且針對第一運算元偵測PER儲存變更事件。若 B_1 欄位為0，則不存取第一運算元。

除了例外檢查以外，亦進行關於CPU是否處於異動執行模式(亦即，異動巢狀結構深度是否為0)之判定(查詢1206)。若CPU並不處於異動執行模式，則儲存選擇之一般暫存器對的內容(步驟1208)。詳言之，將由一般暫存器儲存遮罩指明之一般暫存器對的內容儲存於不能由程式直接地存取之模型相依位置中。

另外，進行關於指令之 B_1 欄位是否為0的判定(查詢1210)。若 B_1 欄位不等於0，則將第一運算元位址置放於異動診斷區塊位址中(步驟1214)，且異動診斷區塊位址有效。另外，自當前PSW之內容設定異動中止PSW(步驟1216)。異動中止PSW之指令位址指明下一循序指令(亦即，在最外部TBEGIN之後的指令)。

此外，判定允許AR修改(A)控制(指令之 I_2 欄位的位元12)之有效值(步驟1218)。有效A控制為針對當前層級及針對所有外部層級之TBEGIN指令中之A控制的邏輯AND。另外，判定允許浮點運算(F)控

制(指令之I₂欄位的位元13)之有效值(步驟1220)。有效F控制為針對當前層級及針對所有外部層級之TBEGIN指令中之F控制的邏輯AND。另外，判定程式中斷篩選控制(PIFC)(指令之I₂欄位的位元14至15)之有效值(步驟1222)。有效PIFC值為針對當前層級及針對所有外部層級之TBEGIN指令中之最高值。

另外，將為1之值添加至異動巢狀結構深度(步驟1224)，且指令用設定條件碼0而完成(步驟1226)。若異動巢狀結構深度自0轉變至1，則CPU進入非受限異動執行模式；否則，CPU保持於非受限異動執行模式。

返回至查詢1210，若B₁等於0，則異動診斷區塊位址無效(步驟1211)，且處理繼續步驟1218。相似地，若CPU處於異動執行模式(查詢1206)，則處理繼續步驟1218。

舉例而言，TBEGIN之執行的所得條件碼包括：

- 0 異動起始成功
- 1 --
- 2 --
- 3 --

舉例而言，程式例外包括：

- 存取(儲存，第一運算元)
- 操作(未安裝異動執行設施)
- 特殊操作
- 規格
- 異動限制(歸因於受限定指令)

在一實施例中，上文所提供之例外檢查可以變化次序而發生。

例外檢查之一特定次序如下：

- 優先權相同於針對一般狀況之程式中斷條件之優先權的例

外。

- 歸因於保留PIFC值之規格例外。
- 歸因於不在雙字組邊界上之第一運算元位址的規格例外。
- 存取例外(當 B_1 欄位為非零時)。
- 歸因於超過最大異動巢狀結構深度之中止。
- 歸因於正常完成之條件碼0。

備註：

1. 當 B_1 欄位為非零時，以下內容適用：

• 當起始最外部異動時，將提供可存取異動診斷區塊(TDB)——即使異動從未中止亦如此。

• 因為TDB之可存取性是否針對巢狀異動被測試係不可預測的，所以應針對任何巢狀TBEGIN指令提供可存取TDB。

• 相比於 B_1 欄位為0時的情形，以下兩者之執行可較慢： B_1 欄位為非零之任何TBEGIN；及針對由 B_1 欄位為非零之最外部TBEGIN起始之異動而發生的任何中止處理。

2. 在一實施例中，若異動並不在異動憑藉TRANSACTION END而正常地結束時中止，則僅恢復經指明以由一般暫存器儲存遮罩儲存之暫存器。在中止時僅恢復由最外部TRANSACTION BEGIN指令之GRSM指明的暫存器。

I_2 欄位應指明提供由異動改變之輸入值的所有暫存器對。因此，若異動中止，則當進入中止處理常式時，輸入暫存器值將恢復為其原始內容。

3. TRANSACTION BEGIN(TBEGIN)指令被預期為繼之以條件分支指令，該條件分支指令將判定異動是否被成功地起始。

4. 若異動歸因於並未引起中斷之條件而中止，則由異動中止PSW指明之指令接收控制(亦即，在最外部TRANSACTION

BEGIN(TBEGIN) 之後的指令)。除了由 TRANSACTION BEGIN(TBEGIN)指令設定之條件碼以外，當異動中止時亦設定條件碼1至3。

因此，在最外部TRANSACTION BEGIN(TBEGIN)指令之後的指令序列應能夠適應所有四個條件碼，即使在此實例中TBEGIN指令僅設定碼0亦如此。

5. 在大多數模型上，既在TRANSACTION BEGIN上又當異動中止時，可藉由指定在一般暫存器儲存遮罩中進行儲存及恢復所需要之暫存器的最小數目而實現改良型效能。

6. 在處於非受限異動執行模式時，程式可呼叫可變更存取暫存器或浮點暫存器(包括浮點控制暫存器)之服務功能。儘管此類服務常式可將已變更暫存器儲存於項上且在結束時恢復該等暫存器，但異動可在常式之正常結束之前中止。若呼叫程式在CPU處於非受限異動執行模式時未爲了保留此等暫存器而做好準備，則呼叫程式可不能容許該等暫存器之服務功能變更。

爲了在處於非受限異動執行模式時防止存取暫存器之非故意變更，程式可將允許AR修改控制(TRANSACTION BEGIN指令之I₂欄位的位元12)設定爲0。相似地，爲了防止浮點暫存器之非故意變更，程式可將允許浮點運算控制(TBEGIN指令之I₂欄位的位元13)設定爲0。

7. 在TRANSACTION BEGIN(TBEGIN)指令之執行期間辨識之程式例外條件經受由任何外部TBEGIN指令設定之有效程式中斷篩選控制。在最外部TBEGIN指令之執行期間辨識之程式例外條件不經受篩選。

8. 爲了以序列化方式更新多個儲存位置，習知碼序列可使用鎖定字組(旗號)。若(a)異動執行用以實施多個儲存位置之更新、(b)程式亦提供「後援」路徑以在異動中止時被調用，且(c)後援路徑使用鎖

定字組，則異動執行路徑亦應針對鎖定之可用性進行測試，且若鎖定不可用，則憑藉 TRANSACTION END 指令來結束異動且分支至後援路徑。此情形確保對序列化資源之一致存取，而不管該等資源是否被異動地更新。

或者，若鎖定不可用，則程式可中止；然而，中止處理可顯著地慢於經由 TEND 簡單地結束異動。

9. 若有效程式中斷篩選控制(PIFC)大於0，則CPU篩選大多數資料例外程式中斷。若有效允許浮點運算(F)控制為0，則資料例外碼(DXC)將由於歸因於資料例外程式例外條件之中止而不在浮點控制暫存器中進行設定。在此情境(篩選適用且有效F控制為0)中，DXC被檢測之唯一位置處於TBEGIN指定之TDB中。若程式之中止處理常式在此類情形中檢測DXC，則一般暫存器B1應為非零，使得設定有效異動診斷區塊位址(TDBA)。

10. 若PER儲存變更或零位址偵測條件針對最外部TBEGIN指令之TBEGIN指定之TDB而存在，且PER事件抑制並不適用，則在指令之執行期間辨識PER事件，因此造成異動立即中止，而不管是否存在任何其他中止條件。

在一實施例中，TBEGIN指令將異動中止位址隱含地設定為在TBEGIN之後的下一循序指令。此位址意欲為條件分支指令，該條件分支指令取決於條件碼(CC)而判定是否進行分支。成功TBEGIN設定CC0，而已中止異動設定CC1、CC2或CC3。

在一實施例中，TBEGIN指令提供指明異動診斷區塊(TDB)之位址的選用儲存運算元，若異動中止，則資訊儲存至該異動診斷區塊(TDB)中。

另外，TBEGIN指令提供包括以下各者之立即運算元：

一般暫存器儲存遮罩(GRSM)，其指示一般暫存器之哪些對將在

異動執行開始時被儲存且在異動中止時被恢復；

允許異動在異動修改存取暫存器時中止之位元(A)；

允許異動在異動嘗試執行浮點指令時中止之位元(F)；及

程式中斷篩選控制(PIFC)，其在異動中止時允許個別異動層級略過程式中斷之實際呈現。

A、F及PIFC控制在各種巢狀結構層級處可不同，且在內部異動層級結束時恢復為先前層級。

此外，TBEGIN(或在另一實施例中，TBEGINC)用以形成異動符記。視情況，可使該符記與藉由TEND指令形成之符記匹配。對於每一TBEGIN(或TBEGINC)指令，作為一實例，符記係由第一運算元位址形成。此符記可獨立於基底暫存器是否為0而形成(不同於僅在基底暫存器為非零時發生的TDB位址設定)。對於用非零基底暫存器而執行之每一TRANSACTION END指令，相似符記係由其儲存運算元形成。若符記並不匹配，則可辨識程式例外以向程式提醒未配對指令。

符記匹配提供意欲藉由確保TEND陳述式與TBEGIN(或TBEGINC)適當地配對而改良軟體可靠性的機制。當TBEGIN指令在特定巢狀結構層級處執行時，符記係由識別異動之此執行個體的儲存運算元位址形成。當執行對應TEND指令時，符記係由指令之儲存運算元位址形成，且CPU比較用於巢狀結構層級之開始符記與結束符記。若符記並不匹配，則辨識例外條件。模型可針對僅某一數目個巢狀結構層級(或不針對巢狀結構層級)實施符記匹配。符記可不涉及儲存運算元位址之所有位元，或位元可經由雜湊或其他方法而組合。符記可藉由TBEGIN指令形成，即使不存取TBEGIN指令之儲存運算元亦如此。

概言之，非受限異動之處理如下：

- 若TND = 0：

- 若 $B_1 \neq 0$ ，則自第一運算元位址設定異動診斷區塊位址。
- 將異動中止PSW設定為下一循序指令位址。
- 將由 I_2 欄位指明之一般暫存器對儲存於模型相依位置中。
- 不能由程式直接地存取。
- 計算有效PIFC、A及F控制
- 有效A = TBEGIN A及任何外部A
- 有效F = TBEGIN F及任何外部F
- 有效PIFC = $\max(\text{TBEGIN PIFC}, \text{任何外部PIFC})$
- 累加異動巢狀結構深度(TND)
- 若TND自0轉變至1，則CPU進入異動執行模式
- 將條件碼設定為0
- 當在TBEGIN之後的指令接收控制時：
 - TBEGIN成功係由CC0指示
 - 已中止異動係由非零CC指示
- 例外：
 - 在巢狀結構深度超過時之中止碼13
 - 在 B_1 欄位為非零且儲存運算元不能被存取以進行儲存操作時之存取例外(各種PIC中之一者)
 - 在TBEGIN指令為執行型別指令之目標時的執行例外(PIC 0003)
 - 在未安裝異動執行設施時之操作例外(PIC 0001)
 - 在以下任一情形時之PIC 0006
 - PIFC無效(為3之值)
 - 第二運算元位址未雙字組對準
 - 在異動執行控制(CR0.8)為0時之PIC 0013十六進位
 - 在受限TX模式下發出時之PIC 0018十六進位

如上文所指示，異動(受限抑或非受限)可由 TRANSACTION END(TEND)指令結束。參看圖13來描述與異動結束(TEND)指令之處理相關的另外細節。執行TEND之CPU(亦即，處理器)執行圖13之邏輯。

參看圖13，最初，基於處理器獲得(例如，提取、接收等等)TEND指令，執行各種例外檢查，且若存在例外(查詢1300)，則處置例外(步驟1302)。舉例而言，若 TRANSACTION END 為執行型別指令之目標，則抑制操作且辨識執行例外；且若異動執行控制(CR0之位元8)為0，則辨識特殊操作例外且抑制操作。再者，若於組態中未安裝異動執行設施，則辨識操作例外且抑制操作。

返回至查詢1300，若未辨識例外，則使異動巢狀結構深度遞減(例如，減1)(步驟1304)。進行關於異動巢狀結構深度在遞減之後是否為0的判定(查詢1306)。若異動巢狀結構深度為0，則交付由異動(及異動巢狀結構內之其他異動(若存在)，此異動為異動巢狀結構之部分)進行之所有儲存存取(步驟1308)。另外，CPU離開異動執行模式(步驟1310)，且指令完成(步驟1312)。

返回至查詢1306，若異動巢狀結構深度不等於0，則 TRANSACTION END指令恰好完成。

若CPU在操作開始時處於異動執行模式，則將條件碼設定為0；否則，將條件碼設定為2。

應注意，有效允許浮點運算(F)控制、允許AR修改(A)控制及程式中斷篩選控制(PIFC)經重設至其在起始正被結束之層級的 TRANSACTION BEGIN指令之前的各別值。另外，在操作完成時執行序列化功能。

在最外部 TRANSACTION END指令完成時辨識之PER指令提取及異動結束事件不會引起異動中止。

在一實例中，TEND指令亦包括基本欄位 B_2 及位移欄位 D_2 ，該兩個欄位經組合(例如，添加)以建立第二運算元位址。在此實例中，可執行符記匹配。舉例而言，當 B_2 為非零時，第二運算元位址之選擇之位元與藉由對應TBEGIN形成之異動符記匹配。若存在失配，則存在例外(例如，PIC 0006)。

除了以上內容以外，異動可由TRANSACTION ABORT指令隱含地或明確地中止。由TABORT或以其他方式使異動中止包括執行數個步驟。一般而言，參看圖14來描述用於中止處理之步驟之實例。若基於中止由TABORT抑或以其他方式起始而存在處理差異，則在以下描述中指示該差異。在一實例中，處理器(例如，CPU)正執行圖14之邏輯。

參看圖14，最初，基於TABORT指令或隱含中止之執行，交付在CPU處於異動執行模式時進行之非異動儲存存取(步驟1400)。捨棄在CPU處於異動執行模式時進行之其他儲存(例如，異動儲存)(步驟1402)。

CPU離開異動執行模式(步驟1404)，且後續儲存非異動地發生。當前PSW係用異動中止PSW之內容取代，惟條件碼係如上文所描述而被設定除外(不同於以下情形，其中若TDBA有效但區塊為不可存取的，則 $CC=1$)(步驟1406)。作為中止處理之部分或在中止處理之後，處理分支至異動中止PSW指定位置以執行動作。在異動為受限異動之一實例中，位置為TBEGINC指令，且動作為彼指令之重新執行；且在異動為非受限異動之另外實例中，位置為在TBEGIN之後的指令，且動作為彼指令之執行，其可為(例如)至中止處理常式之分支。

緊接著，進行關於異動診斷區塊位址是否有效之判定(查詢1408)。當異動診斷區塊位址有效時，將識別中止之原因的診斷資訊及一般暫存器之內容儲存於TBEGIN指定異動診斷區塊中(步驟

1410)。上文參考異動診斷區塊來描述所儲存之TDB欄位及TDB欄位被儲存之條件。

若異動診斷區塊位址有效，但區塊已變得不可存取，則在執行最外部TBEGIN指令之後，不存取該區塊且條件碼1適用。

對於歸因於引起中斷之程式例外條件而中止的異動，儲存程式中斷TDB。

返回至查詢1408，若異動診斷區塊位址無效，則不儲存TBEGIN指定之TDB且條件碼2或3適用，此取決於中止之原因。

除了以上內容以外，亦將異動巢狀結構深度設定為等於0(步驟1412)。另外，恢復經指定以由最外部TBEGIN指令儲存之任何一般暫存器對(步驟1414)。當異動中止時，不恢復未經指明以由最外部TBEGIN指令儲存之一般暫存器對。

另外，執行序列化功能(步驟1416)。序列化功能或操作包括：在發生概念上後續之儲存存取(及相關參考位元及改變位元設定)之前，由CPU完成所有概念上先前之儲存存取(且作為一實例，對於z/Architecture，相關參考位元及改變位元設定)，如由其他CPU及I/O子系統所觀察。序列化實現對儲存體及對儲存體金鑰之所有CPU存取的序列，惟與ART表項及DAT表項提取相關聯之存取除外。

如由處於異動執行模式之CPU所觀察，序列化正常地操作(如上文所描述)。如由其他CPU及I/O子系統所觀察，由於使異動巢狀結構深度遞減至0(正常結束)之TRANSACTION END指令抑或由於正被中止之異動，在CPU處於異動執行模式時執行之序列化操作在CPU離開異動執行模式時發生。

對於以除了由TABORT以外之其他方式起始之中止處理，若異動歸因於引起中斷之例外條件而中止(查詢1418)，則將與中斷相關聯之中斷碼或參數儲存於對應於中斷型別之已指派儲存位置處(步驟

1420)。另外，如上文所設定，將當前PSW儲存至中斷舊PSW中(步驟1422)。此後，或若異動並未歸因於引起中斷之例外條件而中止，則指令以條件碼0而結束。

除了以上內容以外，亦在針對z/Architecture之解譯性執行的一實施例中，當CPU處於異動執行模式且發生將正常地引起攔截碼4、12、44、56、64、68或72之客體條件時，並不發生攔截。實情為，CPU保持於解譯性執行模式，且向客體指示中止條件如下：

- 對於非受限異動，異動歸因於受限定指令而中止(中止碼11)。若偵測到並行PER事件且針對PER而啟動CPU，則程式中斷以中斷碼0280十六進位而發生。

- 對於受限異動，辨識異動限制例外。若偵測到並行PER事件且針對PER而啟動CPU，則程式中斷以中斷碼0298十六進位而發生。

當異動歸因於程式例外條件而中止時，程式中斷篩選可禁止中斷之實際呈現。對於可引起攔截之程式中斷，篩選亦禁止攔截。

參看圖15A至圖15B來描述關於異動處理之另外細節。在一實施例中，此邏輯係由處理器執行。

參看圖15A，最初，經由TRANSACTION BEGIN指令而起始異動(步驟1500)。舉例而言，若異動為受限異動，則使用TBEGINC指令以起始異動。然而，若異動為非受限異動，則使用TBEGIN指令。

在異動之執行期間，若異動中止(查詢1502)，則進行關於異動型別之判定(步驟1504)。此係因為與中止之相關聯之處理取決於異動型別，例如，受限異動或非受限異動。

進行關於異動型別是否為第一型別(諸如，受限異動)之判定(查詢1506)。若異動為受限異動，則基於中止，恢復在TBEGINC指令中指定之一或多個暫存器，且捨棄異動儲存(步驟1508)。另外，進行關於中止是否係歸因於中斷/攔截之判定(查詢1510)。若中止係歸因於中

斷/攔截，則管理中斷/攔截(步驟1512)。舉例而言，若中止係歸因於中斷，則將異動中止PSW(指向TRANSACTION BEGIN且含有指示成功重試之機率的條件碼)儲存為中斷舊PSW。存在針對各種中斷型別(例如，外部中斷、程式中斷、I/O中斷、機器檢查中斷)之分離中斷舊及中斷新PSW。中斷舊PSW指示在中斷時何事項係在控制中。接著，將控制轉移至中斷新PSW(例如，中斷處理常式之位址)。另外，若中止係歸因於攔截，則將異動中止PSW儲存於狀態描述中，且發生攔截，從而使CPU離開解譯性執行模式。然而，若中止並非係歸因於中斷/攔截，則執行在TRANSACTION BEGIN指令處繼續(步驟1516)。

返回至查詢1506，若異動型別為第二型別(諸如，非受限異動)，則視情況將資訊儲存於一或多個異動診斷區塊中(步驟1520)。舉例而言，若異動診斷區塊之有效位址提供於TBEGIN指令中，則診斷資訊儲存於程式指定之TDB(又名，TBEGIN-TDB)中。另外，若異動歸因於某些未篩選程式例外條件而中止，則資訊儲存於程式中斷TDB中；且若異動歸因於某些攔截條件而中止，則資訊儲存於攔截TDB中。

另外，用於非受限異動之中止處理包括恢復TBEGIN指令中指定之一或多個暫存器，及捨棄異動儲存(步驟1522)。另外，進行關於中止是否係歸因於中斷/攔截之判定(查詢1524)。若中止係歸因於中斷/攔截，則如上文，管理中斷/攔截(步驟1526)。然而，若中止並非係歸因於中斷/攔截，則執行在TRANSACTION BEGIN指令之後的指令(例如，在TBEGIN之後的第一指令；亦即，異動中在程式次序上緊接在TBEGIN之後的下一循序指令)處繼續(步驟1530)。

參看圖15B來描述關於在TRANSACTION BEGIN指令之後繼續執行的另外細節。

最初，接收控制之指令檢查異動之條件碼(例如，TBEGIN之條件碼)以判定指令為何正接收控制(步驟1550)。若條件碼指示出在

TRANSACTION BEGIN指令之後的指令歸因於中止而正接收控制(例如，非零條件碼)，則進行關於條件碼是否指示重新執行之另外判定(查詢1556)。舉例而言，若條件碼具有特定值(例如，2)，則指示異動之重新執行且執行彼動作(步驟1558)。然而，若條件指示出重新執行不大可能會成功(例如，條件碼3)(查詢1556)，則採取後援路徑(步驟1560)。

返回至查詢1552，若在TBEGIN指令之後的指令接收除了來自中止以外之控制(例如，零條件碼)，則異動繼續執行(步驟1554)。

上文描述異動處理之一實例，在該實例中，若中止發生，則異動型別用以判定如何執行異動處理。另外，異動型別亦定義中止處理。此情形使異動環境能夠具有不同異動型別。

除了以上內容以外，亦進一步提供在無傳統(粗糙粒度級)序列化(諸如，鎖定)之情況下更新記憶體中之多個不連續物件的有效率方式，該方式提供顯著多處理器效能改良之潛能。亦即，多個不連續物件在無藉由諸如鎖定及旗號之傳統技術提供之更粗糙粒度級儲存存取排序之增強的情況下予以更新。在無費力復原設定的情況下提供理論式執行，且提供受限異動以用於簡單之小使用量更新。

異動執行可用於多種情境中，包括但不限於部分內嵌、理論式處理及鎖定省略。在部分內嵌中，待包括於執行路徑中之部分區包覆於TBEGIN/TEND中。TABORT可包括於其中以在側向出口(side-exit)上回復狀態。對於理論，諸如，以Java，對已取值指標之空值檢查可藉由使用異動而延遲至迴圈邊緣。若指標為空值，則異動可使用TABORT而安全地中止，TABORT包括於TBEGIN/TEND內。

至於鎖定省略，參看圖16A至圖16B及下文所提供之碼片段來描述鎖定省略之使用之一實例。

圖16A描繪複數個佇列元素1602a至1602d之雙向連結清單1600。

新佇列元素1602e將插入至佇列元素之雙向連結清單1600中。每一佇列元素1602a至1602e包括前向指標1604a至1604e及向後指標1606a至1606e。如圖16B所示，爲了將佇列元素1602e添加於佇列元素1602b與佇列元素1602c之間，(1)將向後指標1606e設定爲指向佇列元素1602b、(2)將前向指標1604e設定爲指向佇列元素1602c、(3)將向後指標1606c設定爲指向佇列元素1602e，且(4)將前向指標1604b設定爲指向佇列元素1602e。

對應於圖16A至圖16B之實例碼片段如下：

* R1 - 待插入之新佇列元素的位址。

* R2 - 插入點之位址；新元素插入於由R2指向之元素之前。

NEW	USING	QEL, R1	
CURR	USING	QEL, R2	
	LHI	R15, 10	載入重試計數。
LOOP	TBEGIN	TDB, X'C000'	開始異動(儲存GR 0至3)
	JNZ	ABORTED	非零CC意謂已中止。
	LG	R3, CURR.BWD	指向先前元素。
PREV	USING	QEL, R3	使可定址。
	STG	R1, PREV.FWD	更新先前前向指標。
	STG	R1, CURR.BWD	更新當前向後指標。
	STG	R2, NEW.FWD	更新新前向指標。
	STG	R3, NEW.BWD	更新新向後指標。
	TEND		結束異動。
	...		
ABORTED	JO	NO_RETRY	CC3：非可重試中止。
	JCT	R15, LOOP	重試異動幾分鐘。
	J	NO_RETRY	在10x之後無消息；使

其變得複雜。

在一實例中，若異動用於鎖定省略，但後援路徑使用鎖定，則異動將至少提取鎖定字組以查看鎖定字組是否可用。若另一CPU非異動地存取鎖定，則處理器確保異動中止。

如本文所使用，互換地使用儲存體、中央儲存體、主儲存體、記憶體及主記憶體，除非藉由使用隱含地或明確地另有提及。另外，雖然在一實施例中異動有效地延遲包括延遲將異動儲存交付至主記憶體直至選擇之異動完成；但在另一實施例中，異動有效地延遲包括允許對記憶體之異動更新，但保持舊值且在中止時使記憶體恢復為舊值。

熟習此項技術者應瞭解，一或多項態樣可被體現為一系統、方法或電腦程式產品。因而，一或多項態樣可採取完全硬體實施例、完全軟體實施例(包括韌體、駐留軟體、微碼等等)或組合軟體與硬體態樣之實施例的形式，該等實施例在本文中皆可通稱為「電路」、「模組」或「系統」。此外，一或多項態樣可採取以一或多個電腦可讀媒體體現之電腦程式產品之形式，該一或多個電腦可讀媒體具有體現於其上之電腦可讀程式碼。

可利用一或多個電腦可讀媒體之任何組合。電腦可讀媒體可為電腦可讀儲存媒體。舉例而言，電腦可讀儲存媒體可為但不限於電子、磁性、光學、電磁、紅外線或半導體系統、裝置或器件，或前述各者之任何合適組合。電腦可讀儲存媒體之更特定實例(非詳盡清單)包括以下各者：具有一或多個電線之電連接、攜帶型電腦磁片、硬碟、隨機存取記憶體(RAM)、唯讀記憶體(ROM)、可抹除可程式設計唯讀記憶體(EPROM或快閃記憶體)、光纖、攜帶型光碟唯讀記憶體(CD-ROM)、光學儲存器件、磁性儲存器件，或前述各者之任何合適組合。在此文件之上下文中，電腦可讀儲存媒體可為可含有或儲存供

指令執行系統、裝置或器件使用或結合指令執行系統、裝置或器件而使用之程式的任何有形媒體。

現在參看圖17，在一實例中，電腦程式產品1700包括(例如)一或多個非暫時性電腦可讀儲存媒體1702以在其上儲存電腦可讀程式碼構件或邏輯1704，以提供及促進一或多項態樣。

可使用適當媒體(包括但不限於無線、有線、光纖纜線、RF等等，或前述各者之任何合適組合)傳輸體現於電腦可讀媒體上之程式碼。

可以一或多種程式設計語言之任何組合來撰寫用於進行一或多項態樣之操作的電腦程式碼，該一或多種程式設計語言包括諸如Java、Smalltalk、C++或其類似者之物件導向式程式設計語言，及諸如「C」程式設計語言、組譯器或相似程式設計語言之習知程序性程式設計語言。程式碼可完全地在使用者電腦上執行、部分地在使用者電腦上執行、作為獨立套裝軟體而執行、部分地在使用者電腦上執行且部分地在遠端電腦上執行，或完全地在遠端電腦或伺服器上執行。在後者情境中，遠端電腦可經由任何網路型別(包括區域網路(LAN)或廣域網路(WAN))而連接至使用者之電腦，或可連接至外部電腦(例如，使用網際網路服務提供者而經由網際網路進行連接)。

本文參考方法、裝置(系統)及電腦程式產品之流程圖說明及/或方塊圖來描述一或多項態樣。應理解，可藉由電腦程式指令來實施該等流程圖說明及/或方塊圖之每一方塊及該等流程圖說明及/或方塊圖中之方塊的組合。可將此等電腦程式指令提供至一般用途電腦、特殊用途電腦或其他可程式化資料處理裝置之一處理器以產生一機器，使得經由該電腦或其他可程式化資料處理裝置之該處理器而執行之指令產生用於實施在該或該等流程圖及/或方塊圖方塊中指定之功能/動作之構件。

亦可將此等電腦程式指令儲存於一電腦可讀媒體中，其可指導電腦、其他可程式化資料處理裝置或其他器件以特定方式起作用，使得儲存於該電腦可讀媒體中之指令產生一製品，該製品包括實施在該或該等流程圖及/或方塊圖方塊中指定之功能/動作的指令。

亦可將該等電腦程式指令載入至電腦、其他可程式化資料處理裝置或其他器件上，以使一系列操作步驟在該電腦、其他可程式化裝置或其他器件上執行以產生一電腦實施程序，使得在該電腦或其他可程式化裝置上執行之指令提供用於實施在該或該等流程圖及/或方塊圖方塊中指定之功能/動作的程序。

該等圖中之流程圖及方塊圖說明根據各種實施例的系統、方法及電腦程式產品之可能實施之架構、功能性及操作。就此而言，流程圖或方塊圖中之每一方塊可表示程式碼之一模組、區段或部分，其包含用於實施指定之邏輯功能的一或多個可執行指令。亦應注意，在一些替代性實施中，區塊中所提到之功能可不以諸圖中所提到之次序而發生。舉例而言，取決於所涉及之功能性，連續展示之兩個區塊實際上可實質上並行地執行，或該等區塊有時可以相反次序執行。亦應注意，可藉由執行指定之功能或動作的基於特殊用途硬體之系統或特殊用途硬體及電腦指令之組合來實施方塊圖及/或流程圖說明之每一方塊及方塊圖及/或流程圖說明中之方塊的組合。

除了以上內容以外，一或多項態樣亦可由提供客戶環境之管理的服務提供者提供、部署、管理、服務等等。舉例而言，服務提供者可為一或多個客戶建立、維持、支援等等執行一或多項態樣之電腦程式碼及/或電腦基礎結構。作為回報，作為實例，服務提供者可在訂用及/或收費協議下向客戶收取付款。或者或另外，服務提供者可根據廣告內容至一或多個第三方之銷售而收取付款。

在一實施例中，可部署用於執行一或多個實施例之應用程式。

作為一實例，應用程式之部署包含提供可操作以執行一或多個實施例之電腦基礎結構。

作為一另外態樣，可部署一計算基礎結構，其包含將電腦可讀程式碼整合至計算系統中，其中與該計算系統組合之程式碼能夠執行一或多個實施例。

作為又一態樣，可提供用於整合計算基礎結構之程序，其包含將電腦可讀程式碼整合至電腦系統中。電腦系統包含電腦可讀媒體，其中電腦媒體包含一或多個實施例。與該電腦系統組合之程式碼能夠執行一或多個實施例。

儘管上文描述各種實施例，但此等實施例僅為實例。舉例而言，其他架構之計算環境可用以併入及使用一或多個實施例。另外，可使用不同指令、指令格式、指令欄位及/或指令值。此外，可提供/使用不同、其他及/或額外異動型別。許多變化係可能的。

另外，其他計算環境型別可有益且被使用。作為一實例，適合於儲存及/或執行程式碼之資料處理系統係可用的，其包括直接地或經由系統匯流排間接地耦接至記憶體元件之至少兩個處理器。記憶體元件包括(例如)在程式碼之實際執行期間使用之本端記憶體、大容量儲存體，及快取記憶體，該快取記憶體提供至少某一程式碼之臨時儲存以便減少在執行期間必須自大容量儲存體擷取程式碼的次數。

輸入/輸出或I/O器件(包括但不限於鍵盤、顯示器、指標器件、DASD、磁帶、CD、DVD、隨身碟(thumb drive)及其他記憶體媒體等等)可直接地或經由介入I/O控制器耦接至該系統。網路配接器亦可耦接至系統以使資料處理系統能夠經由介入之私用或公用網路耦接至其他資料處理系統或遠端印表機或儲存器件。數據機、纜線數據機及乙太網路卡僅為可用網路配接器型別中之少數幾種。

參看圖18，描繪實施一或多個實施例之主機電腦系統5000的代

表性組件。代表性主機電腦5000包含與電腦記憶體(亦即，中央儲存體)5002通信之一或多個CPU 5001，以及用於與其他電腦或SAN及其類似者通信之至儲存媒體器件5011及網路5010的I/O介面。CPU 5001符合具有架構化之指令集及架構化之功能性的架構。CPU 5001可具有包括ART後備緩衝器(ALB)5013的存取暫存器轉譯(ART)5012，該存取暫存器轉譯(ART)5012用於選擇待由動態位址轉譯(DAT)5003使用之位址空間從而將程式位址(虛擬位址)變換成記憶體的真實位址。DAT通常包括用於對轉譯進行快取，使得對電腦記憶體5002之區塊之稍後存取並不需要位址轉譯之延遲的轉譯後備緩衝器(TLB)5007。通常，快取記憶體5009用於電腦記憶體5002與處理器5001之間。快取記憶體5009可為階層式快取記憶體，從而具有可用於一個以上CPU之大快取記憶體及大快取記憶體與每一CPU之間的較小之更快(較低層級)快取記憶體。在一些實施中，較低層級快取記憶體經分隔以提供分離之低層級快取記憶體以用於指令提取及資料存取。在一實施例中，對於TX設施，異動診斷區塊(TDB)5100及一或多個緩衝器5101可儲存於快取記憶體5009及記憶體5002中的一或多者中。在一實例中，在TX模式下，資料最初儲存於TX緩衝器中，且當TX模式結束(例如，最外部TEND)時，緩衝器中之資料儲存(交付)至記憶體，或若存在中止，則捨棄緩衝器中的資料。

在一實施例中，經由快取記憶體5009藉由指令提取單元5004自記憶體5002提取指令。指令在指令解碼單元5006中被解碼，且分派(在一些實施例中用其他指令)至指令執行單元5008。通常，使用若干執行單元5008，例如，算術執行單元、浮點執行單元及分支指令執行單元。另外，在TX設施之一實施例中，可使用各種TX控制5110。指令係由執行單元執行，從而在需要時自指令指定之暫存器或記憶體存取運算元。若將自記憶體5002存取(載入或儲存)運算元，則載入/儲存

單元5005通常在正被執行之指令控制下處置存取。指令可在硬體電路中或在內部微碼(韌體)中或由此兩者之組合執行。

根據TX設施之一態樣，處理器5001亦包括PSW 5102(例如，TX及/或中止PSW)、巢狀結構深度5104、TDBA 5106及一或多個控制暫存器5108。

如所提及，電腦系統包括本打(或主)儲存體中之資訊，以及定址、保護，以及參考及改變記錄。定址之一些態樣包括位址之格式、位址空間之概念、各種位址型別，及一位址型別經轉譯至另一位址型別的方式。主儲存體中之一些包括永久指派之儲存位置。主儲存體向系統提供資料之可直接定址之快速存取儲存。資料及程式兩者在其可被處理之前將載入至主儲存體中(自輸入器件)。

主儲存體可包括有時被稱作快取記憶體之一或多個較小較快存取緩衝儲存體。快取記憶體通常與CPU或I/O處理器實體地相關聯。除了效能以外，相異儲存媒體之實體構造及使用的效應通常不能由程式觀察。

可維持用於指令且用於資料運算元的分離快取記憶體。快取記憶體內之資訊維持於被稱作快取記憶體區塊或快取行(或簡稱為行)之整體邊界上的連續位元組中。模型可提供傳回快取行之以位元組為單位之大小的EXTRACT CACHE ATTRIBUTE指令。模型亦可提供PREFETCH DATA及PREFETCH DATA RELATIVE LONG指令，其實現儲存體至資料或指令快取記憶體中之預取或資料自快取記憶體的釋放。

儲存體被視作長水平位元字串。對於大多數操作，對儲存體之存取以自左側至右側序列而行進。將位元字串再分成八個位元之單元。八位元單元被稱作位元組，位元組為所有資訊格式之基本建置區塊。儲存體中之每一位元組位置係藉由唯一非負整數識別，該非負整

數為彼位元組位置之位址，或簡稱為位元組位址。鄰近位元組位置具有連續位址，其在左側以0開始且以自左側至右側序列而行進。位址為無正負號二進位整數，且為24、31或64個位元。

在儲存體與CPU或通道子系統之間一次一個位元組或位元組群組而傳輸資訊。除非另有指定，否則在(例如)z/Architecture中，儲存體中之位元組群組係由該群組之最左側位元組定址。群組中位元組之數目係由待執行之操作暗示抑或明確地指定。當用於CPU操作中時，位元組群組被稱作欄位。在每一位元組群組內，在(例如)z/Architecture中，位元係以自左側至右側序列被編號。在z/Architecture中，最左側位元有時被稱作「高階」位元，且最右側位元被稱作「低階」位元。然而，位元號碼並非儲存位址。可僅定址位元組。為了對儲存體中之位元組的個別位元進行操作，存取整個位元組。位元組中之位元自左側至右側被編號為0至7(在(例如)z/Architecture中)。位址中之位元對於24位元位址可被編號為8至31或40至63，或對於31位元位址可被編號為1至31或33至63；位址中之位元對於64位元位址被編號為0至63。在一實例中，位元8至31及1至31應用於處於為32個位元寬之位置(例如，暫存器)中的位址，而位元40至63及33至63應用於處於64位元寬位置中的位址。在多個位元組之任何其他固定長度格式內，構成該格式之位元自0開始進行連續地編號。出於錯誤偵測之目的，且較佳地出於校正起見，可用每一位元組或用一位元組群組來傳輸一或多個檢查位元。此類檢查位元係由機器自動地產生且不能直接地受到程式控制。以位元組之數目來表達儲存容量。當儲存運算元欄位之長度由指令之操作碼暗示時，欄位據稱具有固定長度，該固定長度可為一個、兩個、四個、八個或十六個位元組。較大欄位可針對一些指令進行暗示。當儲存運算元欄位之長度未被暗示而是被明確地陳述時，欄位據稱具有可變長度。可變長度運算元長度可以一位元組之增量(或用一

些指令，以兩個位元組之倍數或其他倍數)發生變化。當將資訊置放於儲存體中時，包括於指定欄位中的僅彼等位元組位置之內容被取代，即使至儲存體之實體路徑的寬度可大於正被儲存之欄位的長度亦如此。

資訊之某些單元將在儲存體中之整體邊界上。當儲存位址為以位元組為單位之單元之長度的倍數時，邊界對於資訊單元被稱作整體。向整體邊界上之2、4、8、16及32位元組之欄位提供特殊名稱。半字組為二位元組邊界上之兩個連續位元組的群組，且為指令之基本建置區塊。字組為四位元組邊界上之四個連續位元組的群組。雙字組為八位元組邊界上之八個連續位元組的群組。四倍字組為16位元組邊界上之16個連續位元組的群組。八倍字組為32位元組邊界上之32個連續位元組的群組。當儲存位址指明半字組、字組、雙字組、四倍字組及八倍字組時，位址之二進位表示分別含有一個、兩個、三個、四個或五個最右側零位元。指令將處於二位元組整體邊界上。大多數指令之儲存運算元並不具有邊界對準要求。

在實施用於指令及資料運算元之分離快取記憶體之器件上，若將程式儲存至供隨後提取指令之快取行中，則可經歷顯著延遲，而不管儲存是否變更隨後被提取的指令。

在一實例中，實施例可由軟體(有時指代已授權內碼、韌體、微碼、毫碼、皮碼及其類似者，前述各者中任一者將與一或多個實施例一致)實踐。參看圖18，體現一或多項態樣之軟體程式碼可由主機系統5000之處理器5001自諸如CD-ROM碟機、磁帶機或硬碟機之長期儲存媒體器件5011存取。軟體程式碼可體現於多種已知媒體中任一者上以供資料處理系統使用，諸如，磁片、硬碟機或CD-ROM。程式碼可散佈於此類媒體上，或可自電腦記憶體5002散佈至使用者或經由網路5010自一電腦系統之儲存體散佈至其他電腦系統以供此類其他系統之

使用者使用。

軟體程式碼包括控制各種電腦組件與一或多個應用程式之功能及互動的作業系統。程式碼正常地自儲存媒體器件5011傳呼至程式碼可用於由處理器5001進行處理的相對較高速度電腦儲存體5002。用於將軟體程式碼體現於記憶體中、實體媒體上及/或經由網路散佈軟體碼的技術及方法係熟知的，且在本文中將不予以進一步論述。程式碼在建立及儲存於有形媒體(包括但不限於電子記憶體模組(RAM)、快閃記憶體、光碟(CD)、DVD、磁帶及其類似者)上常常被稱作「電腦程式產品」。電腦程式產品媒體通常可由較佳地在電腦系統中之處理電路讀取以供處理電路執行。

圖19說明可實踐一或多個實施例之代表性工作站或伺服器硬體系統。圖19之系統5020包含包括選用周邊器件之代表性基本電腦系統5021，諸如，個人電腦、工作站或伺服器。基本電腦系統5021包括一或多個處理器5026，及用以根據已知技術在處理器5026與系統5021之其他組件之間進行連接及啓用處理器5026與其他組件之間的通信之匯流排。舉例而言，匯流排將處理器5026連接至記憶體5025及長期儲存體5027，長期儲存體5027可包括硬碟機(包括(例如)磁性媒體、CD、DVD及快閃記憶體中任一者)或磁帶機。系統5021可能亦包括使用者介面配接器，其經由匯流排將微處理器5026連接至一或多個介面器件，諸如，鍵盤5024、滑鼠5023、印表機/掃描器5030及/或其他介面器件，該等其他介面器件可為諸如觸敏式螢幕、數位化鍵入板(entry pad)等等之任何使用者介面器件。匯流排亦經由顯示配接器將諸如LCD螢幕或監視器之顯示器件5022連接至微處理器5026。

系統5021可憑藉能夠與網路5029通信5028之網路配接器而與其他電腦或電腦之網路通信。實例網路配接器為通信通道、符記環、乙太網路或數據機。或者，系統5021可使用諸如CDPD(蜂巢式數位封包

資料)卡之無線介面進行通信。系統5021可與區域網路(LAN)或廣域網路(WAN)中之此類其他電腦相關聯，或系統5021可為具有另一電腦之用戶端/伺服器配置中的用戶端等等。在此項技術中知曉所有此等組態以及適當通信硬體及軟體。

圖20說明可實踐一或多個實施例之資料處理網路5040。資料處理網路5040可包括複數個個別網路，諸如，無線網路及有線網路，該等網路中每一者可包括複數個個別工作站5041、5042、5043、5044。另外，熟習此項技術者應瞭解，可包括一或多個LAN，其中LAN可包含耦接至主機處理器之複數個智慧型工作站。

仍參看圖20，網路亦可包括大型主機電腦或伺服器，諸如，閘道器電腦(用戶端伺服器5046)或應用程式伺服器(可存取資料儲存庫且亦可直接地自工作站5045進行存取的遠端伺服器5048)。閘道器電腦5046充當至每一個別網路中之進入點。當將一個網路連接協定連接至另一網路連接協定時需要閘道器。閘道器5046可較佳地憑藉通信鏈路耦接至另一網路(例如，網際網路5047)。閘道器5046亦可使用通信鏈路直接地耦接至一或多個工作站5041、5042、5043、5044。閘道器電腦可利用可購自International Business Machines Corporation之IBM eServer System z伺服器予以實施。

並行地參看圖19及圖20，可體現一或多項態樣的軟體程式設計碼5031可由系統5020之處理器5026自諸如CD-ROM碟機或硬碟機之長期儲存媒體5027存取。軟體程式設計碼可體現於多種已知媒體中任一者上以供資料處理系統使用，諸如，磁片、硬碟機或CD-ROM。程式碼可散佈於此類媒體上，或可自記憶體散佈至使用者5050、5051或經由網路自一電腦系統之儲存體散佈至其他電腦系統以供此類其他系統之使用者使用。

或者，程式設計碼可體現於記憶體5025中，且由處理器5026使

用處理器匯流排而存取。此類程式設計碼包括控制各種電腦組件與一或多個應用程式5032之功能及互動的作業系統。程式碼正常地自儲存媒體5027傳呼至程式碼可用於由處理器5026進行處理的高速度記憶體5025。用於將軟體程式設計碼體現於記憶體中、實體媒體上及/或經由網路散佈軟體碼的技術及方法係熟知的，且在本文中將不予以進一步論述。程式碼在建立及儲存於有形媒體(包括但不限於電子記憶體模組(RAM)、快閃記憶體、光碟(CD)、DVD、磁帶及其類似者)上時常常被稱作「電腦程式產品」。電腦程式產品媒體通常可由較佳地在電腦系統中之處理電路讀取以供處理電路執行。

最易於可用於處理器之快取記憶體(相比於處理器之其他快取記憶體通常較快且較小)為最低(L1或層級1)快取記憶體，且主儲存體(主記憶體)為最高層級快取記憶體(若存在3個層級，則為L3)。最低層級快取記憶體常常被劃分成保持待執行之機器指令的指令快取記憶體(I快取記憶體)，及保持資料運算元之資料快取記憶體(D快取記憶體)。

參看圖21，描繪用於處理器5026之例示性處理器實施例。通常，快取記憶體5053之一或多個層級用以緩衝記憶體區塊以便改良處理器效能。快取記憶體5053為保持最可能使用之記憶體資料之快取行的高速度緩衝器。典型快取行為64、128或256個位元組之記憶體資料。分離快取記憶體除了用於快取資料以外亦常常用於快取指令。快取記憶體連貫性(記憶體及快取記憶體中行之複本的不同步)常常由此項技術中熟知之各種「窺探」演算法提供。處理器系統之主記憶體儲存體5025常常被稱作快取記憶體。在具有快取記憶體5053之4個層級的處理器系統中，主儲存體5025有時被稱作層級5(L5)快取記憶體，此係因為處理器系統通常較快且僅保持可用於電腦系統之非揮發性儲存體(DASD、磁帶等等)之部分。主儲存體5025「快取」由作業系統分頁進及分頁出主儲存體5025之資料頁。

程式計數器(指令計數器)5061追蹤待執行之當前指令的位址。z/Architecture處理器中之程式計數器為64個位元，且可被截斷為31或24個位元以支援先前定址極限。程式計數器通常體現於電腦之PSW(程式狀態字組)中，使得程式計數器在上下文切換期間持續。因此，具有程式計數器值的在進展中之程式可由(例如)作業系統中斷(自程式環境至作業系統環境之上下文切換)。程式之PSW在程式未處於作用中時維持程式計數器值，且作業系統之程式計數器(在PSW中)在作業系統正執行時被使用。通常，使程式計數器累加等於當前指令之位元組之數目的量。RISC(精簡指令集計算)指令通常具有固定長度，而CISC(複雜指令集計算)指令通常具有可變長度。IBM z/Architecture之指令為具有2、4或6個位元組之長度的CISC指令。舉例而言，程式計數器5061係藉由上下文切換操作抑或分支指令之分支採取操作而修改。在上下文切換操作中，當前程式計數器值連同關於正被執行之程式的其他狀態資訊(諸如，條件碼)一起儲存於程式狀態字組中，且載入新程式計數器值從而指向待執行之新程式模組的指令。分支採取操作經執行以便藉由將分支指令之結果載入至程式計數器5061中來准許程式作出決策或在程式內循環。

通常，指令提取單元5055用以以處理器5026之名義來提取指令。提取單元提取「下一循序指令」、分支採取指令之目標指令，抑或在上下文切換之後的程式之第一指令。現代指令提取單元常常使用預取技術以基於可能使用已預取指令之可能性來理論式地預取指令。舉例而言，提取單元可提取包括下一循序指令的指令之16個位元組及另外循序指令之額外位元組。

已提取指令接著由處理器5026執行。在一實施例中，已提取指令傳遞至提取單元之分派單元5056。該分派單元解碼指令，且將關於已解碼指令之資訊轉遞至適當單元5057、5058、5060。執行單元5057

通常將自指令提取單元5055接收關於已解碼算術指令之資訊，且將根據指令之作業碼而對運算元執行算術運算。運算元較佳地自記憶體5025、架構化暫存器5059抑或自正被執行之指令的立即欄位提供至執行單元5057。執行之結果在被儲存時儲存於記憶體5025、暫存器5059中或其他機器硬體(諸如，控制暫存器、PSW暫存器及其類似者)中。

虛擬位址係使用動態位址轉譯5062且視情況使用存取暫存器轉譯5063而轉譯成真實位址。

處理器5026通常具有一或多個單元5057、5058、5060以用於執行指令之功能。參看圖22A，執行單元5057可憑藉介接邏輯5071而與架構化一般暫存器5059、解碼/分派單元5056、載入儲存單元5060及其他處理器單元5065通信。執行單元5057可使用若干暫存器電路5067、5068、5069以保持算術邏輯單元(ALU)5066將進行運算的資訊。ALU執行諸如加法、減法、乘法及除法之算術運算，以及諸如與、或及互斥或(XOR)、旋轉及移位之邏輯功能。較佳地，ALU支援設計相依之特殊化運算。舉例而言，其他電路可提供包括條件碼及復原支援邏輯的其他架構化設施5072。通常，ALU運算之結果保持於輸出暫存器電路5070中，輸出暫存器電路5070可將結果轉遞至多種其他處理功能。存在處理器單元之許多配置，但本描述僅意欲提供一實施例之代表性理解。

ADD指令(例如)將在具有算術及邏輯功能性之執行單元5057中執行，而浮點指令(例如)將在具有特殊化浮點能力之浮點執行中執行。較佳地，執行單元藉由對由指令識別之運算元執行作業碼定義功能而對運算元進行運算。舉例而言，ADD指令可由執行單元5057對在由該指令之暫存器欄位識別之兩個暫存器5059中發現的運算元執行。

執行單元5057對兩個運算元執行算術加法，且將結果儲存於第三運算元中，其中第三運算元可為第三暫存器或兩個來源暫存器中之

一者。執行單元較佳地利用算術邏輯單元(ALU)5066，ALU 5066能夠執行諸如移位、旋轉、與、或及XOR之多種邏輯功能，以及包括加法、減法、乘法、除法中任一者之多種代數功能。一些ALU 5066經設計成用於純量運算且一些ALU 5066經設計成用於浮點。取決於架構，資料可為位元組由大到小(Big Endian)(其中最低有效位元組處於最高位元組位址)或位元組由小到大(Little Endian)(其中最低有效位元組處於最低位元組位址)。IBM z/Architecture為位元組由大到小。取決於架構，帶正負號欄位可為正負號及量值、1之補數，或2之補數。2之補數有利之處在於：ALU無需設計減法能力，此係因為2之補數中的負值或正值僅需要在ALU內之加法。舉例而言，常常以速記法來描述數字，其中12位元欄位定義4,096位元組區塊之位址，且通常被描述為4千位元組區塊。

參看圖22B，用於執行分支指令之分支指令資訊通常發送至分支單元5058，分支單元5058常常使用諸如分支歷史表5082之分支預測演算法以在其他條件操作完成之前預測分支之結果。當前分支指令之目標將被提取，且在條件操作完成之前被理論式地執行。當完成條件操作時，理論式執行之分支指令基於條件操作之條件及所理論結果而被完成抑或捨棄。典型分支指令可測試條件碼，且在條件碼滿足分支指令之分支要求時分支至目標位址，目標位址可基於(例如)包括在暫存器欄位或指令之立即欄位中發現之數字的若干數字予以計算。分支單元5058可使用ALU 5074，ALU 5074具有複數個輸入暫存器電路5075、5076、5077及輸出暫存器電路5080。舉例而言，分支單元5058可與一般暫存器5059、解碼分派單元5056或其他電路5073通信5081。

舉例而言，指令群組之執行可由於多種原因而中斷，該等原因包括由作業系統起始之上下文切換、造成上下文切換之程式例外或錯誤、造成上下文切換之I/O中斷信號，或複數個程式之多執行緒活動

(在多執行緒化環境中)。較佳地，上下文切換動作儲存關於當前執行程式之狀態資訊，且接著載入關於正被調用之另一程式的狀態資訊。舉例而言，狀態資訊可儲存於硬體暫存器中或記憶體中。狀態資訊較佳地包含指向待執行之下一指令的程式計數器值、條件碼、記憶體轉譯資訊，及架構化暫存器內容。上下文切換活動可由硬體電路、應用程式、作業系統程式或韌體碼(微碼、皮碼或已授權內碼(LIC))單獨地或組合地演練。

處理器根據指令定義方法而存取運算元。指令可使用指令之部分之值來提供立即運算元，可提供明確地指向一般用途暫存器或特殊用途暫存器(例如，浮點暫存器)之一或多個暫存器欄位。指令可利用由作業碼欄位識別之所暗示暫存器作為運算元。指令可將記憶體位置用於運算元。如由z/Architecture長位移設施所例示，運算元之記憶體位置可由暫存器、立即欄位或暫存器與立即欄位之組合提供，其中指令定義基底暫存器、索引暫存器及立即欄位(位移欄位)，前述三者加在一起以提供(例如)運算元在記憶體中之位址。本文中之位置通常暗示主記憶體(主儲存體)中之位置，除非另有指示。

參看圖22C，處理器使用載入/儲存單元5060來存取儲存體。載入/儲存單元5060可藉由獲得目標運算元在記憶體5053中之位址且將運算元載入於暫存器5059或另一記憶體5053之位置而執行載入操作，或可藉由獲得目標運算元在記憶體5053中之位址且將自暫存器5059或另一記憶體5053之位置獲得的資料儲存於記憶體5053中之目標運算元位置中而執行儲存操作。載入/儲存單元5060可為理論式，且可以相對於指令序列無序的序列存取記憶體，然而，載入/儲存單元5060對程式維持指令被按次序執行的外觀。載入/儲存單元5060可與一般暫存器5059、解碼/分派單元5056、快取記憶體/記憶體介面5053或其他元件5083通信5084，且包含各種暫存器電路5086、5087、5088及5089、

ALU 5085及控制邏輯5090以計算儲存位址且提供管線定序以使操作按次序。一些操作可無序，但載入/儲存單元提供使無序操作對於程式看來像是已按次序被執行的功能性，如此項技術中所熟知。

較佳地，應用程式所「查看」之位址常常被稱作虛擬位址。虛擬位址有時被稱作「邏輯位址」及「有效位址」。此等虛擬位址為虛擬之處在於：其係藉由多種動態位址轉譯(DAT)技術中之一者重新引導至實體記憶體位置，該等技術包括但不限於簡單地用偏移值來加前置詞於虛擬位址、經由一或多個轉譯表轉譯虛擬位址，轉譯表較佳單獨地或組合地至少包含區段表及頁表，較佳地，區段表具有指向頁表之項。在z/Architecture中，提供轉譯階層，包括區第一表、區第二表、區第三表、區段表及選用頁表。常常藉由利用轉譯後備緩衝器(TLB)來改良位址轉譯之效能，該TLB包含將虛擬位址映射至關聯實體記憶體位置之項。當DAT使用轉移譯來轉譯虛擬位址時建立該等項。虛擬位址之後續使用接著可利用快速TLB之項而非緩慢循序轉譯表存取。TLB內容可由包括LRU(最近最少使用)之多種取代演算法管理。

在處理器為多處理器系統之處理器的狀況下，每一處理器具有使諸如I/O、快取記憶體、TLB及記憶體之共用資源出於一致性而保持連鎖的職責。通常，「窺探」技術將用來維持快取記憶體一致性。在窺探環境中，可將每一快取行標示為處於共用狀態、獨佔狀態、已改變狀態、無效狀態及其類似者中任一者以便促進共用。

舉例而言，I/O單元5054(圖21)向處理器提供用於附接至包括磁帶、碟片、印表機、顯示器及網路之周邊器件的構件。I/O單元常常由軟體驅動程式呈現給電腦程式。在諸如可購自IBM®之System z的大型主機中，通道配接器及開放系統配接器為在作業系統與周邊器件之間提供通信的大型主機之I/O單元。

另外，其他計算環境型別可受益於一或多項態樣。作為一實例，環境可包括模擬器(例如，軟體或其他模擬機制)，其中特定架構(包括(例如)指令執行、架構化功能(諸如，位址轉譯)及架構化暫存器)或其子集被模擬(例如，在具有處理器及記憶體之原生電腦系統上)。在此類環境中，模擬器之一或多個模擬功能可實施一或多個實例，即使執行該模擬器之電腦可具有不同於正被模擬之能力的架構亦如此。作為一實例，在模擬模式下，解碼正被模擬之特定指令或操作，且建置適當模擬功能以實施個別指令或操作。

在一模擬環境中，主機電腦包括(例如)：記憶體，其儲存指令及資料；指令提取單元，其自記憶體提取指令，且視情況提供用於已提取指令之本端緩衝；指令解碼單元，其接收已提取指令且判定已被提取之指令型別；及指令執行單元，其執行該等指令。執行可包括將資料自記憶體載入至暫存器中；將資料自暫存器儲存回至記憶體；或執行某一算術或邏輯運算型別，如由解碼單元所判定。在一實例中，以軟體來實施每一單元。舉例而言，由該等單元執行之操作被實施為模擬器軟體內之一或多個次常式。

更特定而言，在大型主機中，架構化機器指令常常憑藉編譯器應用程式供程式設計師(通常現今為「C」程式設計師)使用。儲存於儲存媒體中之此等指令可原生地在z/Architecture IBM[®]伺服器中或者在執行其他架構之機器中執行。指令可在現有及未來IBM[®]大型主機伺服器中且在IBM[®]之其他機器(例如，Power System伺服器及System x伺服器)上進行模擬。指令可在使用由IBM[®]、Intel[®]、AMD及其他製造商製造之硬體的各種各樣之機器上執行Linux的機器中執行。除了在依據z/Architecture之彼硬體上的執行以外，亦可使用Linux以及數個使用由Hercules、UMX或FSI(Fundamental Software公司)進行之模擬的機器，在該等機器處執行通常處於模擬模式。在模擬模式下，模

擬軟體係由原生處理器執行以對模擬處理器之架構進行模擬。

原生處理器通常執行模擬軟體，該模擬軟體包含韌體或原生作業系統以執行模擬處理器之模擬。模擬軟體負責提取及執行模擬處理器架構之指令。模擬軟體維持模擬程式計數器以追蹤指令邊界。模擬軟體一次可提取一或多個模擬機器指令，且將該一或多個模擬機器指令轉換成原生機器指令之對應群組以供原生處理器執行。此等已轉換指令可被快取，使得可實現較快轉換。然而，模擬軟體仍維持模擬處理器架構之架構規則以便確信針對模擬處理器撰寫之作業系統及應用程式正確地操作。此外，模擬軟體將提供由模擬處理器結構識別之資源，使得經設計成在模擬處理器上執行之作業系統或應用程式可在具有模擬軟體之原生處理器上執行，該等資源包括但不限於控制暫存器、一般用途暫存器、浮點暫存器、包括(例如)區段表及頁表之動態位址轉譯函式、中斷機制、上下文切換機制、當日時間(TOD)時鐘，及至I/O子系統之架構化介面。

解碼正被模擬之指定指令，且呼叫次常式以執行個別指令之功能。對模擬處理器之功能進行模擬的模擬軟體功能係以(例如)以下各者予以實施：「C」次常式或驅動程式，或提供用於特定硬體之驅動程式的某一其他方法，此在理解較佳實施例之描述之後將在熟習此項技術者之技藝內。包括但不限於以下各者之各種軟體及硬體模擬專利說明多種已知方法來達成針對不同機器架構化之指令格式的模擬以用於可用於熟習此項技術者的目標機器：Beausoleil等人之名為「Multiprocessor for Hardware Emulation」之美國專利證書第5,551,013號；及Scalzi等人之名為「Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor」的美國專利證書第6,009,261號；及Davidian等人之名為「Decoding Guest Instruction to Directly Access Emulation Routines

that Emulate the Guest Instructions」的美國專利證書第5,574,873號；及Gorishek等人之名為「Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System」的美國專利證書第6,308,255號；及Lethin等人之名為「Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method」的美國專利證書第6,463,582號；及Eric Traut之名為「Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions」的美國專利證書第5,790,825號，前述專利證書中每一者之全文係據此以引用方式併入本文中；及許多其他專利證書。

在圖23中，提供模擬主機電腦系統5092之實例，其模擬主機架構之主機電腦系統5000'。在模擬主機電腦系統5092中，主機處理器(CPU)5091為模擬主機處理器(或虛擬主機處理器)，且包含具有不同於主機電腦5000'之處理器5091之原生指令集架構的原生指令集架構之模擬處理器5093。模擬主機電腦系統5092具有對於模擬處理器5093可存取的記憶體5094。在實例實施例中，記憶體5094經分割成主機電腦記憶體5096部分及模擬常式5097部分。主機電腦記憶體5096根據主機電腦架構可用於模擬主機電腦5092的程式。模擬處理器5093執行不同於模擬處理器5091之架構的架構之架構化之指令集的原生指令，該等原生指令自模擬常式記憶體5097獲得；且可藉由使用在序列與存取/解碼常式中獲得之一或多個指令來自主機電腦記憶體5096中之程式存取主機指令以供執行，序列與存取/解碼常式可解碼所存取之主機指令以判定用於模擬所存取之主機指令之功能的原生指令執行常式。舉例而言，針對主機電腦系統5000'之架構定義之其他設施可藉由架構化之設施常式來模擬，包括作為一般用途暫存器、控制暫存器、動

態位址轉譯及I/O子系統支援及處理器快取記憶體之此類設施。模擬常式亦可利用在模擬處理器5093中可用之功能(諸如，一般暫存器及虛擬位址之動態轉譯)以改良模擬常式的效能。專用硬體及卸載引擎亦可經提供以在模擬主機電腦5000'之功能中協助處理器5093。

本文所使用之術語僅出於描述特定實施例之目的，且並不意欲為限制性的。如本文所使用，除非上下文另有清楚指示，否則單數形式「一」及「該」意欲亦包括複數形式。應進一步理解，術語「包含」在用於本說明書中時指定存在所敘述特徵、整數、步驟、操作、元件及/或組件，但並不排除存在或添加一或多種其他特徵、整數、步驟、操作、元件、組件及/或其群組。

以下申請專利範圍中之所有構件或步驟附加功能元件的對應結構、材料、動作及等效者(若有)意欲包括用於結合如特定主張之其他所主張元件執行功能的任何結構、材料或動作。出於說明及描述之目的已呈現一或多個實施例之描述，但該描述並不意欲為詳盡的或將本發明限於所揭示之形式。許多修改及變化對於一般熟習此項技術者將顯而易見。選擇及描述實施例以便最佳地解釋各種態樣及實務應用，且使其他一般熟習此項技術者能夠理解具有如適合於所涵蓋之特定用途的各種修改之各種實施例。

【符號說明】

100	計算環境
102	中央處理器複合體
106	輸入/輸出(I/O)器件
108	控制單元
110	中央處理器
112	分割區
114	邏輯分割區超管理器

115	輸入/輸出子系統
200	異動開始(TBEGIN)指令
202	作業碼欄位
204	基本欄位(B ₁)
206	位移欄位(D ₁)
208	立即欄位(I ₂)
210	暫存器儲存遮罩(GRSM)
212	允許AR修改(A)
214	允許浮點運算(F)
216	程式中斷篩選控制(PIFC)
300	異動開始受限(TBEGINC)指令
302	作業碼欄位
304	基本欄位(B ₁)
306	位移欄位(D ₁)
308	立即欄位(I ₂)
310	一般暫存器儲存遮罩(GRSM)
312	允許AR修改(A)
400	異動結束(TEND)指令
402	作業碼欄位
500	異動中止(TABORT)指令
502	作業碼欄位
504	基本欄位(B ₂)
506	位移欄位(D ₂)
600	第一TBEGIN
601	最外部異動
602	異動開始(TBEGIN)

604	異動開始(TBEGIN)
606	異動結束(TEND)
608	最內部異動
610	異動結束(TEND)
612	交付
700	NONTRANSACTIONAL STORE指令
702a	作業碼欄位
702b	作業碼欄位
704	暫存器欄位(R_1)
706	索引欄位(X_2)
708	基本欄位(B_2)
710	第一位移欄位(DL_2)
712	第二位移欄位(DH_2)
800	EXTRACT TRANSACTION NESTING DEPTH指令
802	作業碼欄位
804	暫存器欄位 R_1
900	異動診斷區塊
902	格式
904	旗標
906	異動巢狀結構深度(TND)
908	異動中止碼(TAC)
910	衝突符記
912	已中止異動指令位址(ATIA)
914	例外存取識別(EAID)
916	資料例外碼(DXC)
918	程式中斷識別(PIID)

920	轉譯例外識別(TEID)
922	中斷事件位址
924	模型相依診斷資訊
930	一般暫存器
1600	雙向連結清單
1602a	佇列元素
1602b	佇列元素
1602c	佇列元素
1602d	佇列元素
1602e	佇列元素
1604a	前向指標
1604b	前向指標
1604c	前向指標
1604d	前向指標
1604e	前向指標
1606a	向後指標
1606b	向後指標
1606c	向後指標
1606d	向後指標
1606e	向後指標
1700	電腦程式產品
1702	非暫時性電腦可讀儲存媒體
1704	電腦可讀程式碼構件或邏輯
5000	主機電腦系統
5000'	主機電腦系統
5001	中央處理單元(CPU)

5002	電腦記憶體/中央儲存體
5003	動態位址轉譯(DAT)
5004	指令提取單元
5005	載入/儲存單元
5006	指令解碼單元
5007	轉譯後備緩衝器(TLB)
5008	執行單元
5009	快取記憶體
5010	網路
5011	儲存媒體器件
5012	存取暫存器轉譯(ART)
5013	ART後備緩衝器(ALB)
5020	系統
5021	基礎電腦系統
5022	顯示器件
5023	滑鼠
5024	鍵盤
5025	記憶體/主儲存體
5026	處理器
5027	長期儲存體
5028	通信
5029	網路
5030	印表機/掃描器
5031	軟體程式設計碼
5032	應用程式
5040	資料處理網路

5041	工作站
5042	工作站
5043	工作站
5044	工作站
5045	工作站
5046	閘道器電腦/用戶端伺服器
5047	網際網路
5048	遠端伺服器
5050	使用者
5051	使用者
5053	快取記憶體
5054	I/O單元
5055	指令提取單元
5056	解碼/分派單元
5057	執行單元
5058	分支單元
5059	一般暫存器
5060	載入儲存單元
5061	程式計數器/指令計數器
5062	動態位址轉譯
5063	存取暫存器轉譯
5065	其他處理器單元
5066	算術邏輯單元(ALU)
5067	暫存器電路
5068	暫存器電路
5069	暫存器電路

5070	輸出暫存器電路
5071	介接邏輯
5072	設施
5073	其他電路
5074	算術邏輯單元(ALU)
5075	輸入暫存器電路
5076	輸入暫存器電路
5077	輸入暫存器電路
5080	輸出暫存器電路
5081	通信
5082	分支歷史表
5083	其他元件
5084	通信
5085	算術邏輯單元(ALU)
5086	暫存器電路
5087	暫存器電路
5088	暫存器電路
5089	暫存器電路
5090	控制邏輯
5091	主機處理器(CPU)
5092	模擬主機電腦系統
5093	模擬處理器
5094	記憶體
5096	主機電腦記憶體
5097	模擬常式
5100	異動診斷區塊(TDB)

5101	緩衝器
5102	程式狀態字組(PSW)
5104	巢狀結構深度
5106	TDB位址(TDBA)
5108	控制暫存器
5110	TX控制

申請專利範圍

1. 一種用於控制一異動在一計算環境中之執行之電腦程式產品，該電腦程式產品包含：
 - 一電腦可讀儲存媒體，其可由一處理電路讀取且儲存指令以供該處理電路執行來執行一方法，該方法包含：
 - 由一處理器經由一異動開始指令在一異動模式下起始一異動之執行，該異動開始指令開始一第一異動型別或一第二異動型別中任一者，該異動包含複數個待執行指令；
 - 基於處於異動模式，有效地延遲將異動儲存交付至主記憶體直至一選擇之異動完成為止；
 - 基於該異動中止且基於該異動為一第一異動型別，在該異動開始指令處繼續執行；及
 - 基於該異動中止且基於該異動為一第二異動型別，在程式次序上緊接在該異動開始指令之後的一下一循序指令處繼續執行。
2. 如請求項1之電腦程式產品，其中該第一異動型別為一受限異動，該受限異動具有與其相關聯之一或多個限定，且該異動開始指令為一異動開始受限指令，且其中該第二異動型別為一非受限異動。
3. 如請求項1之電腦程式產品，其中該異動為該第二異動型別，該第二型別為一非受限異動，且其中該繼續執行包含：
 - 藉由在該異動開始指令之後的該下一循序指令檢查該異動之一條件碼以判定該指令是否正自一中止接收控制；及
 - 基於該指令自該中止接收控制，執行處理以處置該中止。
4. 如請求項3之電腦程式產品，其中該執行處理包含：基於該條件

碼而採取動作，其中基於為一個值之一條件碼，該動作包括重新執行該異動，且基於為另一值之一條件碼，該動作包括分支至一後援路徑。

5. 如請求項1之電腦程式產品，其中該方法進一步包含：

執行該異動開始指令，該執行將一條件碼設定為一第一值；
及

執行該異動之一中止，該執行將該條件碼設定為一第二值，其中該第二值指示在該異動開始指令之後的該下一循序指令正自該中止接收控制。

6. 如請求項1之電腦程式產品，其中該異動為該第二異動型別，該第二型別為一非受限異動，且其中該方法進一步包含基於該異動中止，將診斷資訊儲存於由該異動開始指令指定之一異動診斷區塊中。

7. 如請求項1之電腦程式產品，其中該方法進一步包含基於該中止，恢復在該異動開始指令中指明之一或多個暫存器且捨棄異動儲存。

8. 如請求項1之電腦程式產品，其中該方法進一步包含：

判定出該異動已中止；

基於判定出該異動已中止，判定該中止係歸因於一攔截抑或中斷；及

基於該中止係歸因於不同於一攔截或中斷之原因，取決於該異動為該第一型別或該第二型別而分別執行該在該異動開始指令處繼續執行或該在該下一循序指令處繼續執行。

9. 如請求項1之電腦程式產品，其中該異動為一異動巢狀結構之部分，且其中該方法進一步包含由該處理器執行一擷取巢狀結構深度指令以獲得該異動巢狀結構之一巢狀結構深度之一指示。

10. 如請求項9之電腦程式產品，其中該異動為該異動巢狀結構之一最外部異動。
11. 一種用於控制一異動在一計算環境中之執行之電腦系統，該電腦系統包含：
 - 一記憶體；及
 - 一處理器，其與該記憶體通信，其中該電腦系統經組態以執行一方法，該方法包含：
 - 由一處理器經由一異動開始指令在一異動模式下起始一異動之執行，該異動開始指令開始一第一型別異動或一第二型別異動中任一者，該異動包含複數個待執行指令；
 - 基於處於異動模式，有效地延遲將異動儲存交付至主記憶體直至一選擇之異動完成為止；
 - 基於該異動中止且基於該異動為一第一異動型別，在該異動開始指令處繼續執行；及
 - 基於該異動中止且基於該異動為一第二異動型別，在程式次序上緊接在該異動開始指令之後的一下一循序指令處繼續執行。
12. 如請求項11之電腦系統，其中該第一異動型別為一受限異動，該受限異動具有與其相關聯之一或多個限定，且該異動開始指令為一異動開始受限指令，且其中該第二異動型別為一非受限異動。
13. 如請求項11之電腦系統，其中該異動為該第二異動型別，該第二型別為一非受限異動，且其中該繼續執行包含：
 - 藉由在該異動開始指令之後的該下一循序指令檢查該異動之一條件碼以判定該指令是否正自一中止接收控制；及
 - 基於該指令自該中止接收控制，執行處理以處置該中止。

14. 如請求項13之電腦系統，其中該執行處理包含基於該條件碼而採取動作，其中基於為一個值之一條件碼，該動作包括重新執行該異動，且基於為另一值之一條件碼，該動作包括分支至一後援路徑。
15. 如請求項11之電腦系統，其中該異動為該第二異動型別，該第二型別為一非受限異動，且其中該方法進一步包含基於該異動中止，將診斷資訊儲存於由該異動開始指令指定之一異動診斷區塊中。
16. 如請求項11之電腦系統，其中該方法進一步包含基於該中止，恢復在該異動開始指令中指明之一或多個暫存器且捨棄異動儲存。
17. 如請求項11之電腦系統，其中該方法進一步包含：
 - 判定出該異動已中止；
 - 基於判定出該異動已中止，判定該中止係歸因於一攔截抑或中斷；及
 - 基於該中止係歸因於不同於一攔截或中斷之原因，取決於該異動為該第一型別或該第二型別而分別執行該在該異動開始指令處繼續執行或該在該下一循序指令處繼續執行。
18. 一種控制一異動在一計算環境中之執行之方法，該方法包含：
 - 由一處理器經由一異動開始指令在一異動模式下起始一異動之執行，該異動開始指令開始一第一型別異動或一第二型別異動中任一者，該異動包含複數個待執行指令；
 - 基於處於異動模式，有效地延遲將異動儲存交付至主記憶體直至一選擇之異動完成為止；
 - 基於該異動中止且基於該異動為一第一異動型別，在該異動開始指令處繼續執行；及

基於該異動中止且基於該異動為一第二異動型別，在程式次序上緊接在該異動開始指令之後的一下一循序指令處繼續執行。

19. 如請求項18之方法，其中該異動為該第二異動型別，該第二型別為一非受限異動，且其中該繼續執行包含：

藉由在該異動開始指令之後的該下一循序指令檢查該異動之一條件碼以判定該指令是否正自一中止接收控制；及

基於該指令自該中止接收控制，執行處理以處置該中止。

20. 如請求項19之方法，其中該執行處理包含基於該條件碼而採取動作，其中基於為一個值之一條件碼，該動作包括重新執行該異動，且基於為另一值之一條件碼，該動作包括分支至一後援路徑。

圖式

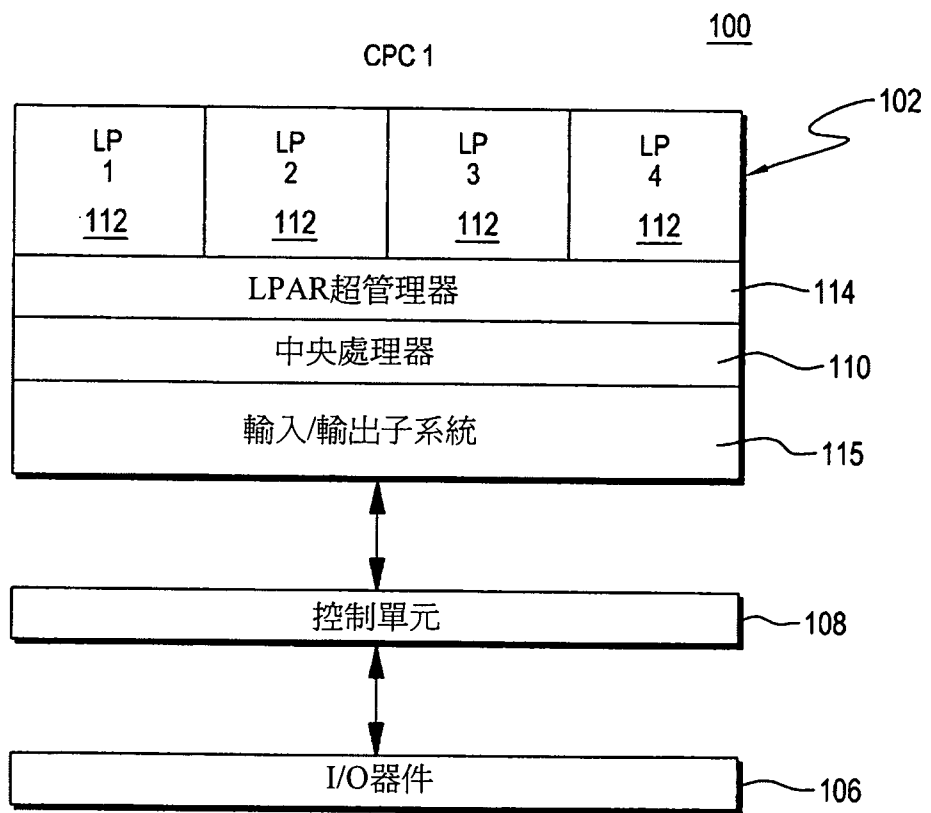


圖1

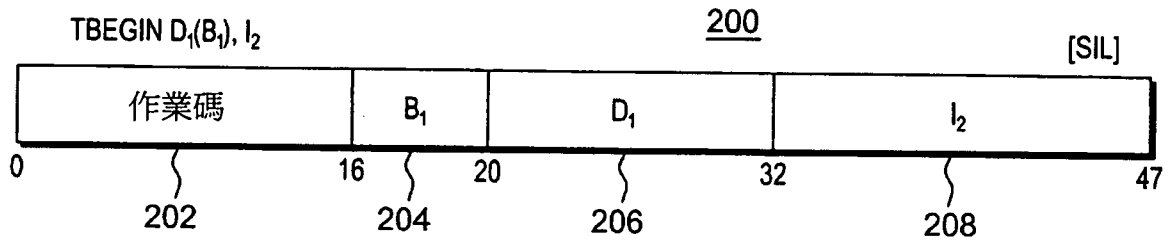


圖2A

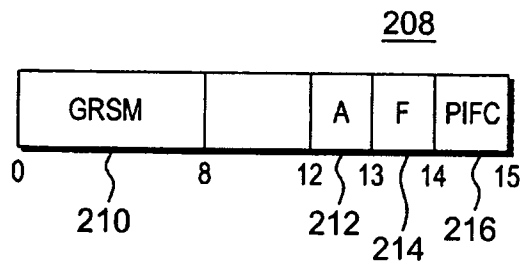


圖2B

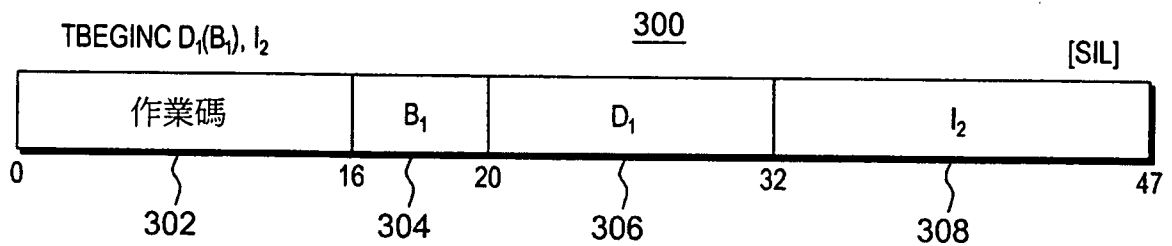


圖3A

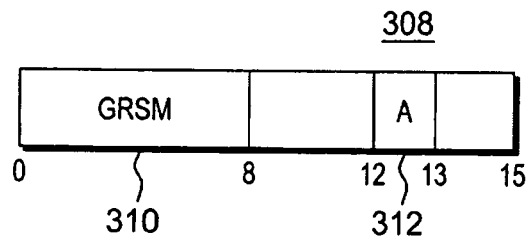


圖3B

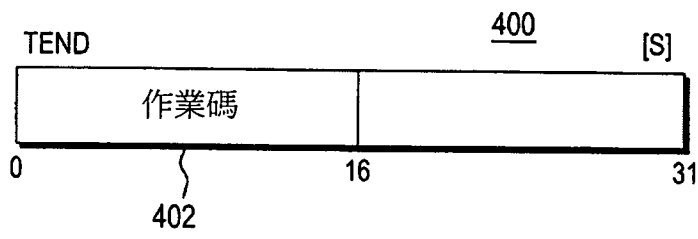


圖4

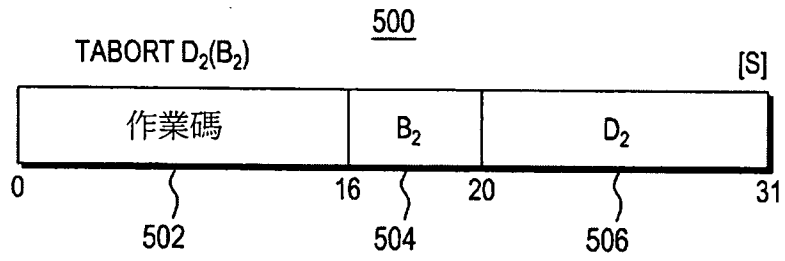


圖5

巢狀異動

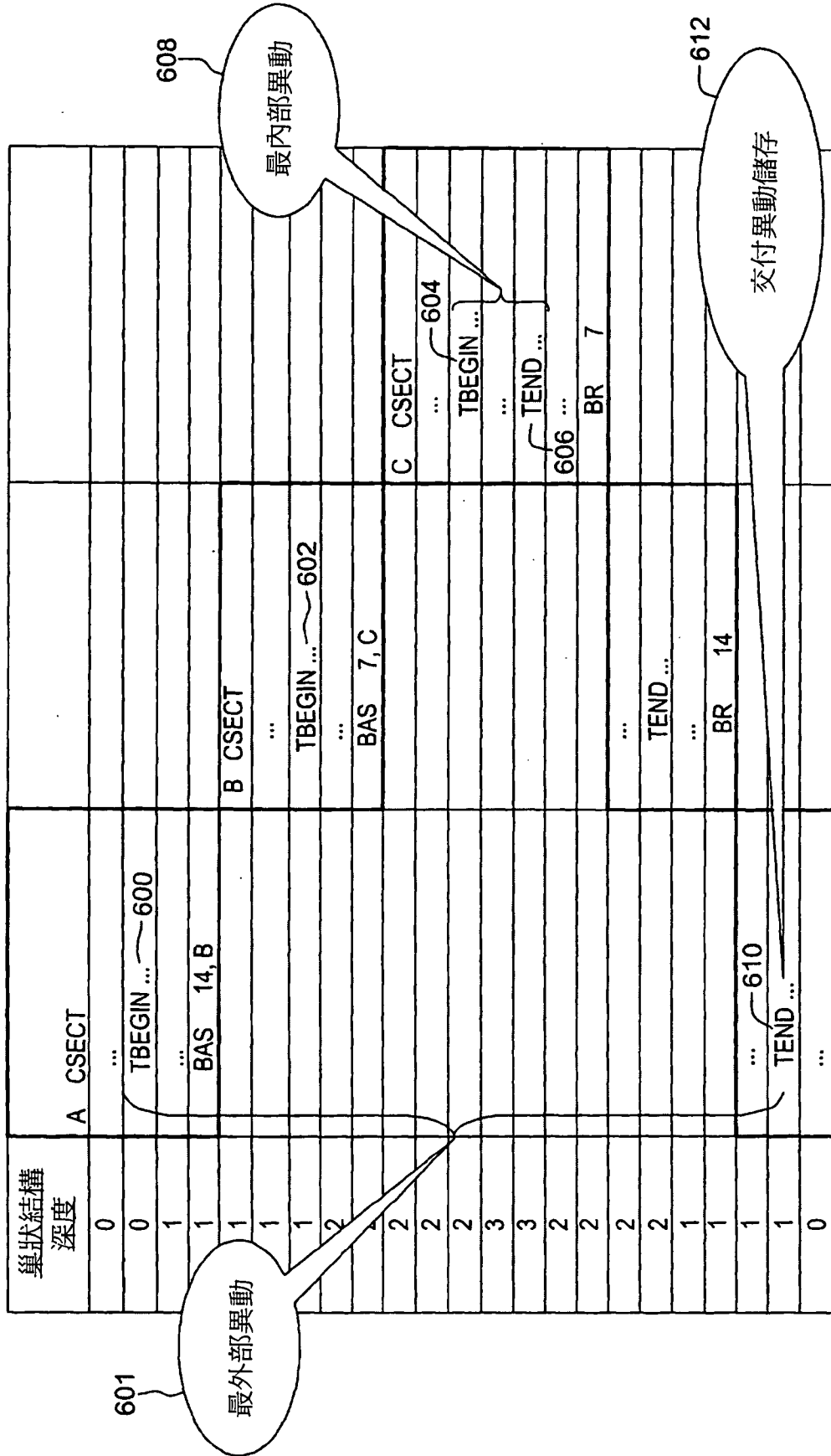


圖6

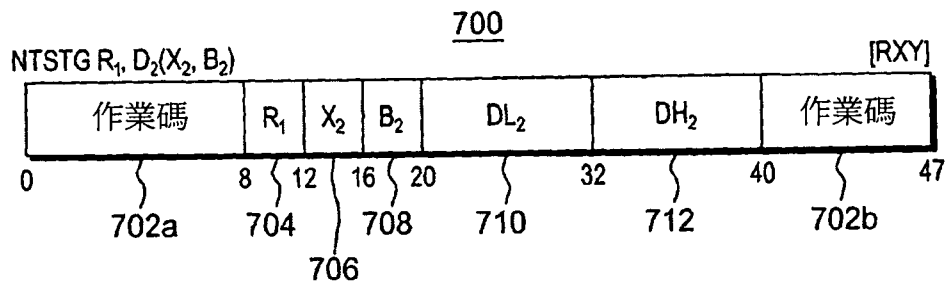


圖7

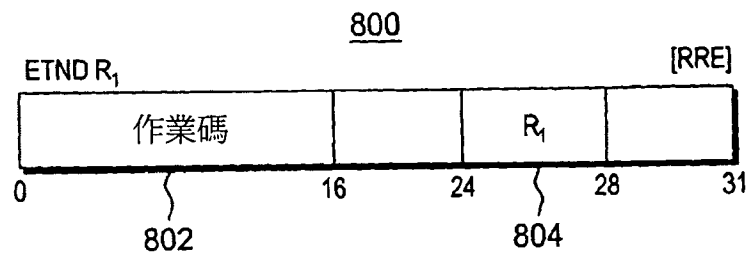


圖8

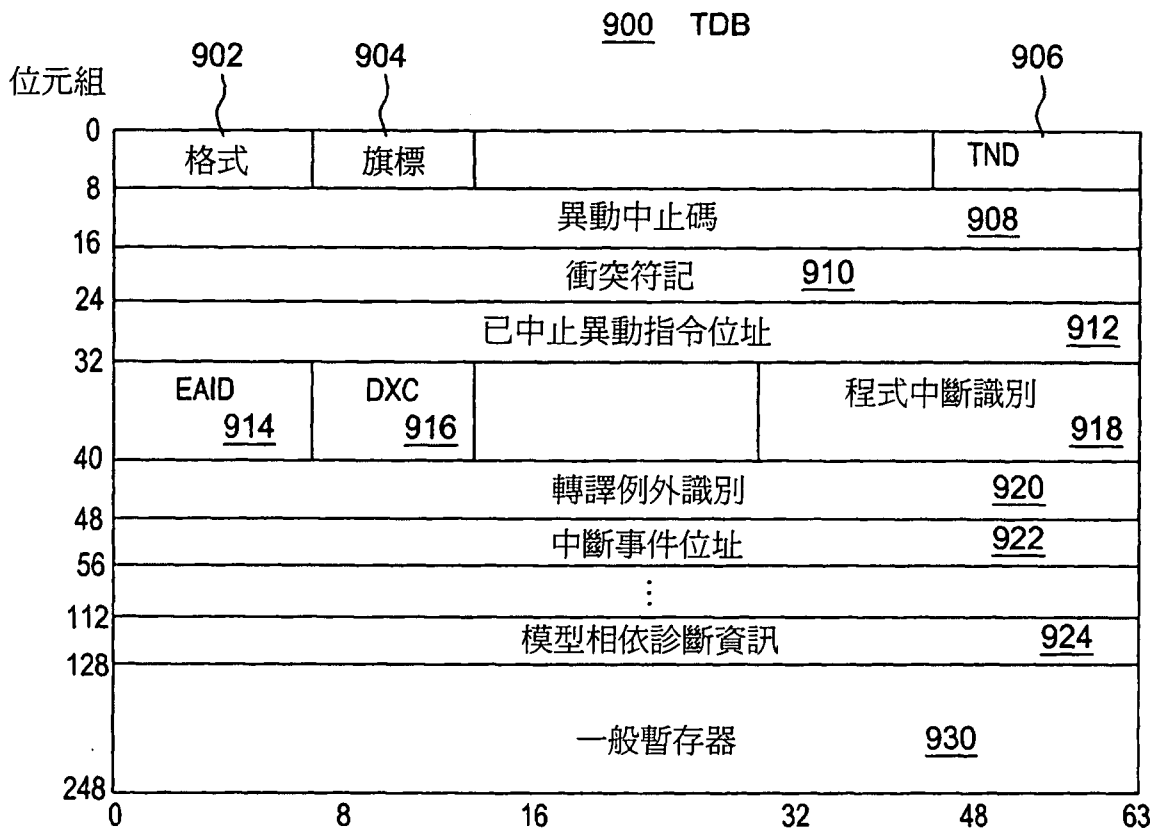


圖9

碼	中止之原因	對應條件碼 (CC)設定
2	外部中斷	2
4	程式中斷(未篩選)	2 或 3+
5	機器檢查中斷	2
6	I/O中斷	2
7	提取溢位	2 或 3
8	儲存溢位	2 或 3
9	提取衝突	2
10	儲存衝突	2
11	受限定指令	3
12	程式例外條件(已篩選)	3
13	巢狀結構深度超過	3
14	快取提取相關	2 或 3
15	快取儲存相關	2 或 3
16	快取其他	2 或 3
255	雜項條件	2 或 3
>255	TABORT指令	2 或 3
‡	不能被判定；不儲存異動診斷區塊(TDB)	1
解釋： + 條件碼係基於中斷碼 ‡ 此情形發生在異動中止但在最外部TBEGIN指令之成功執行之後TDB已變成不可存取時。不儲存TBEGIN指定之TDB，且將條件碼設定為1。		

圖10

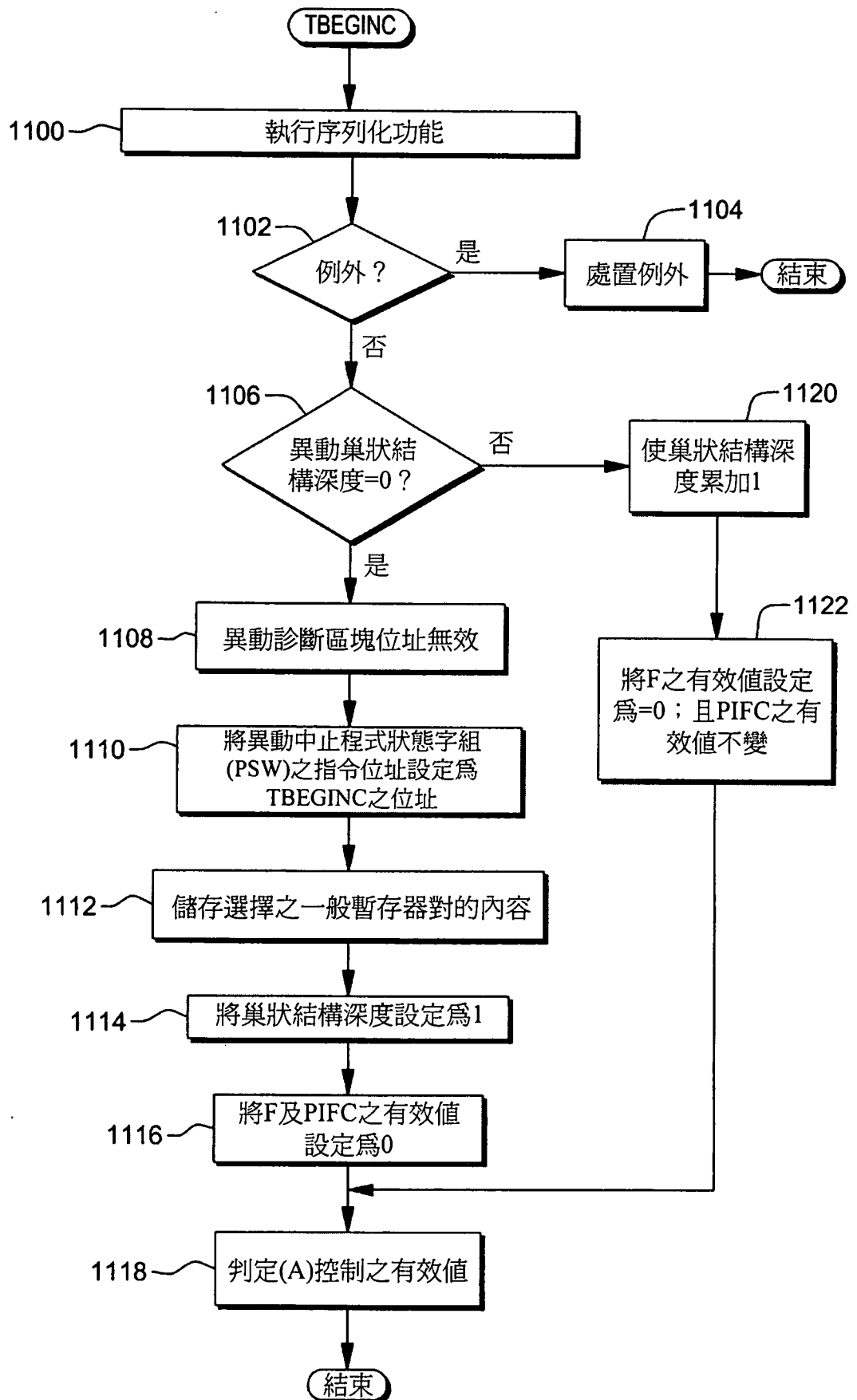


圖11

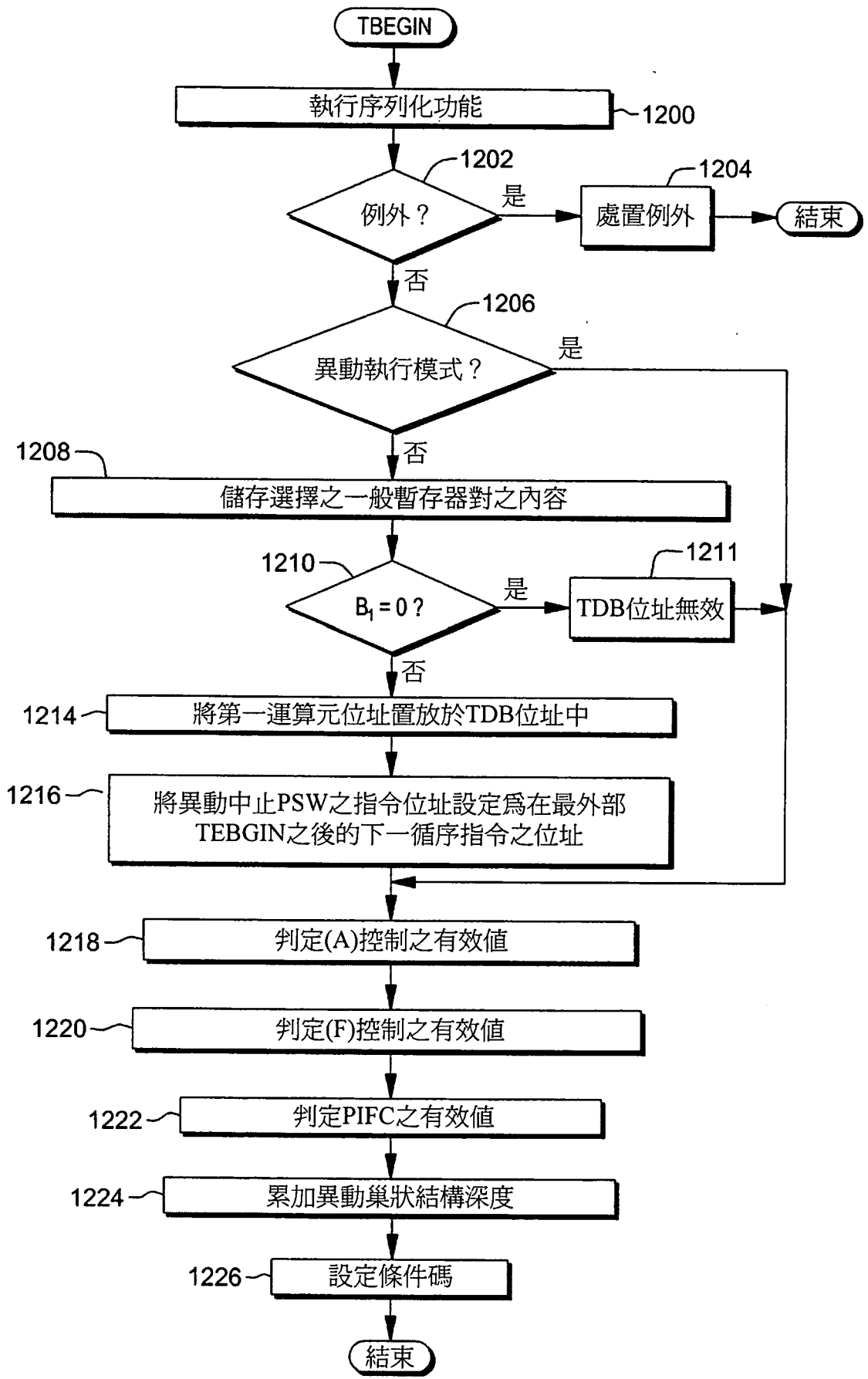


圖12

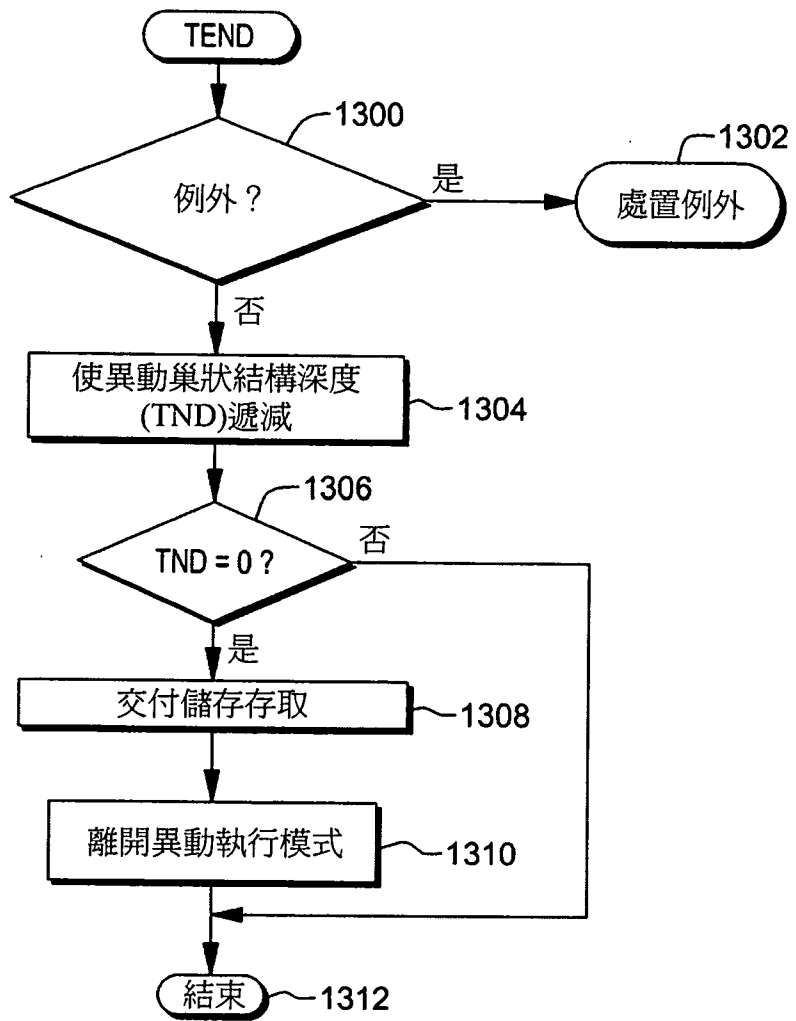


圖13

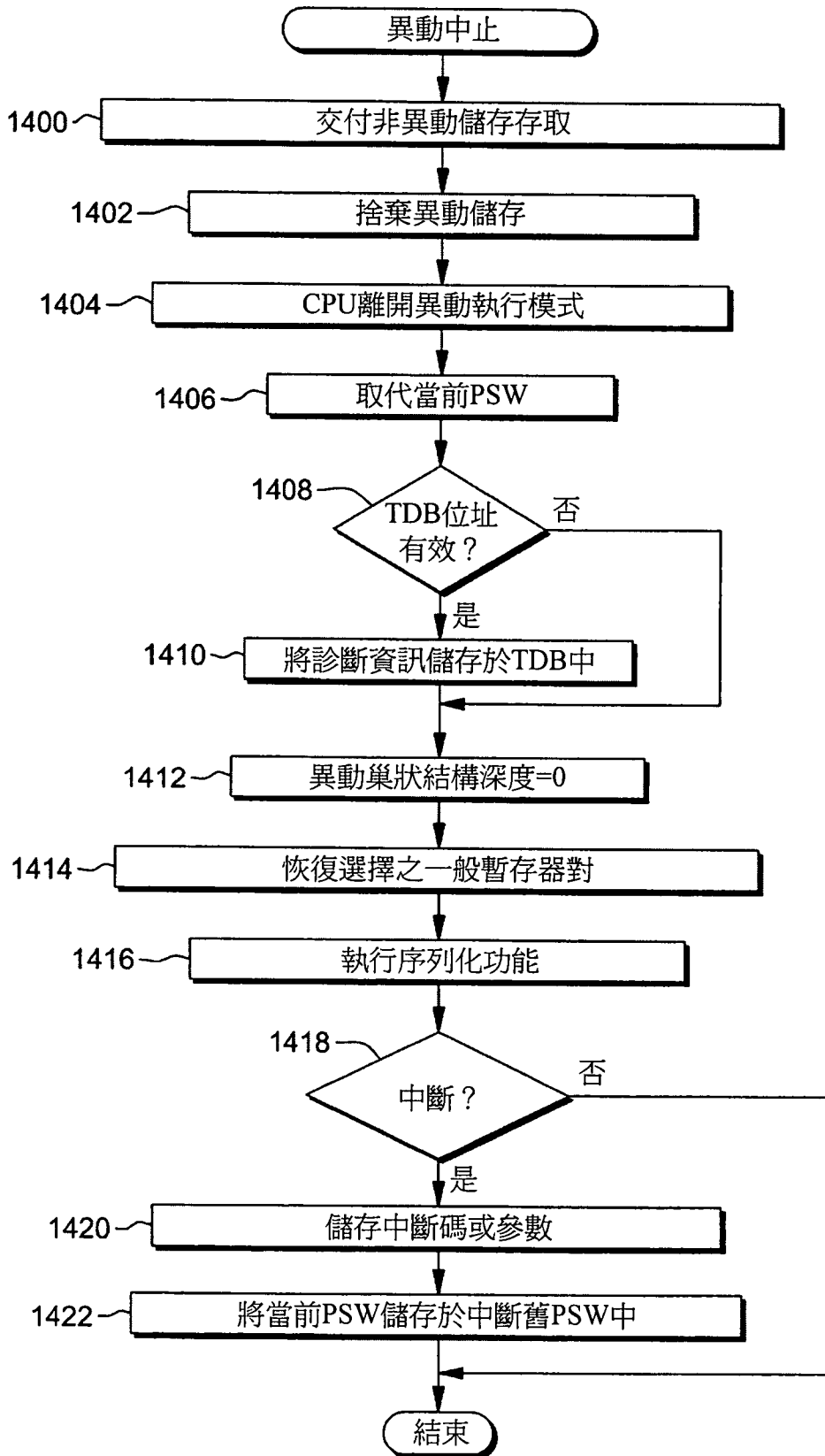


圖14

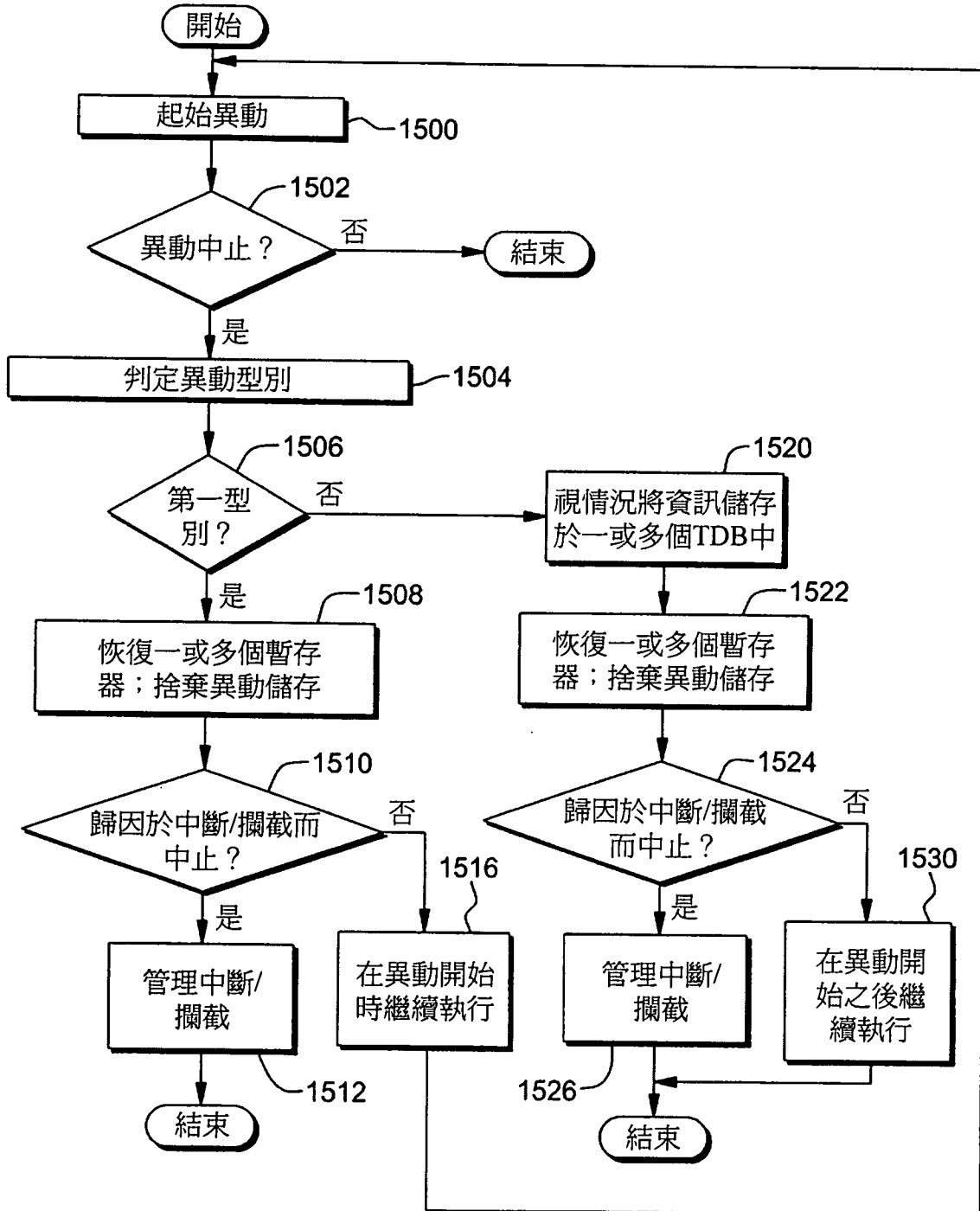


圖15A

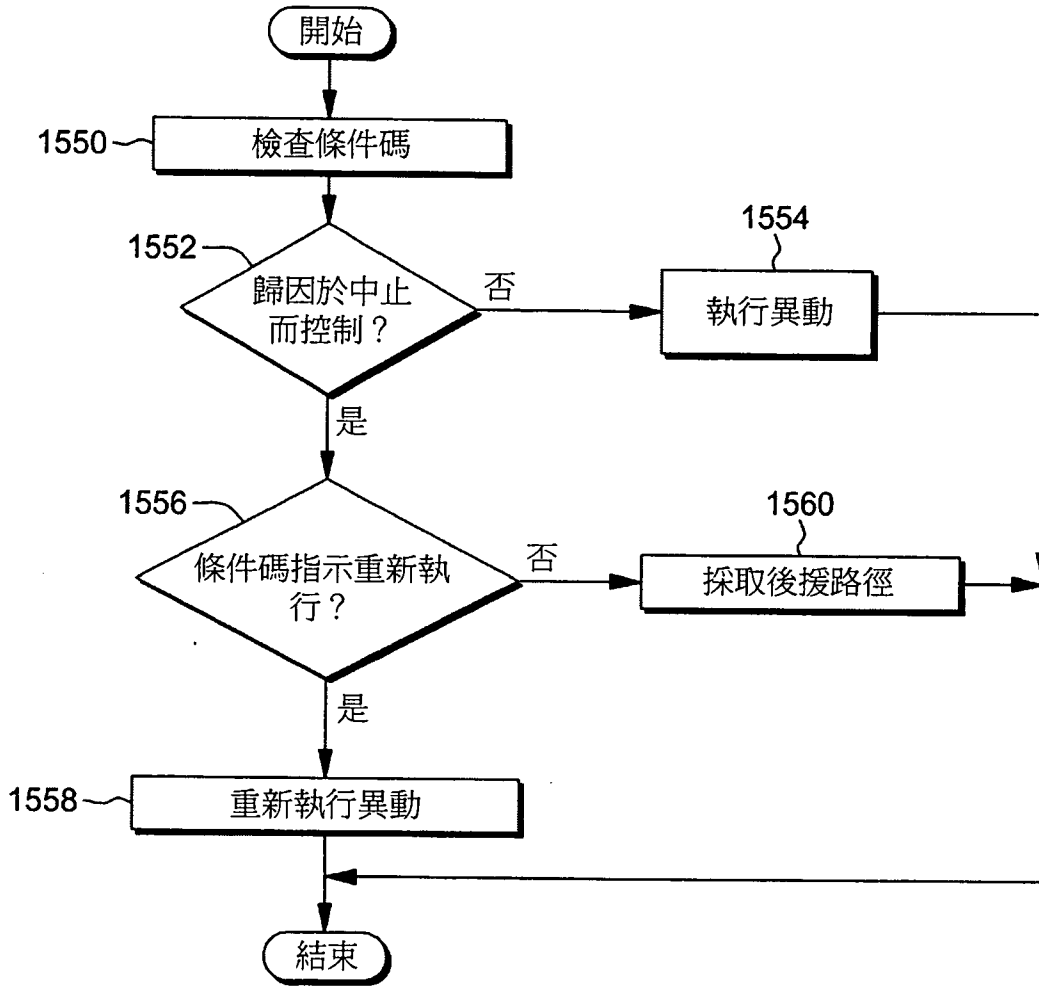


圖15B

將元素插入至雙向連結清單中(之前)

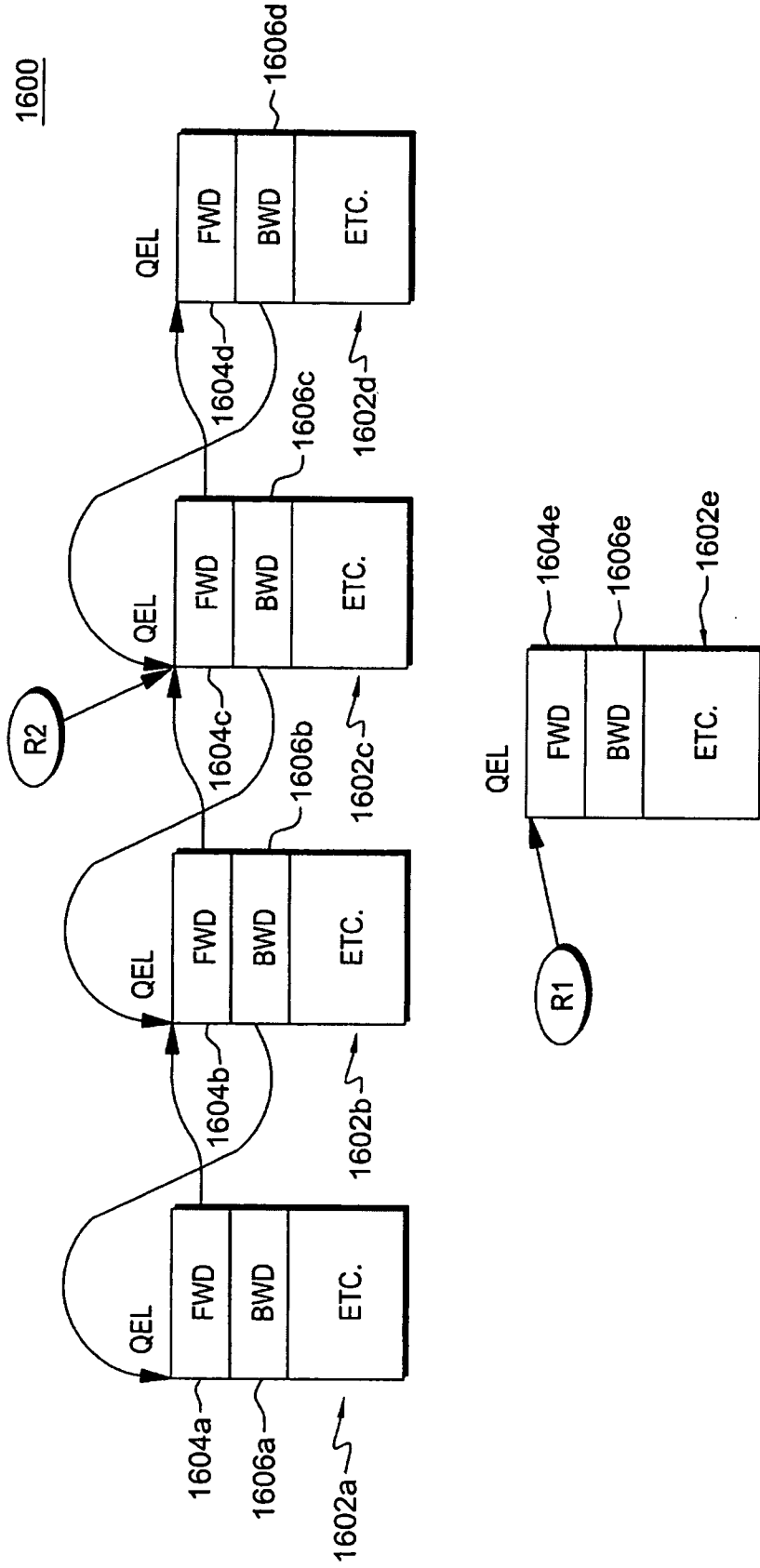


圖16A

將元素插入至雙向連結清單中(之後)

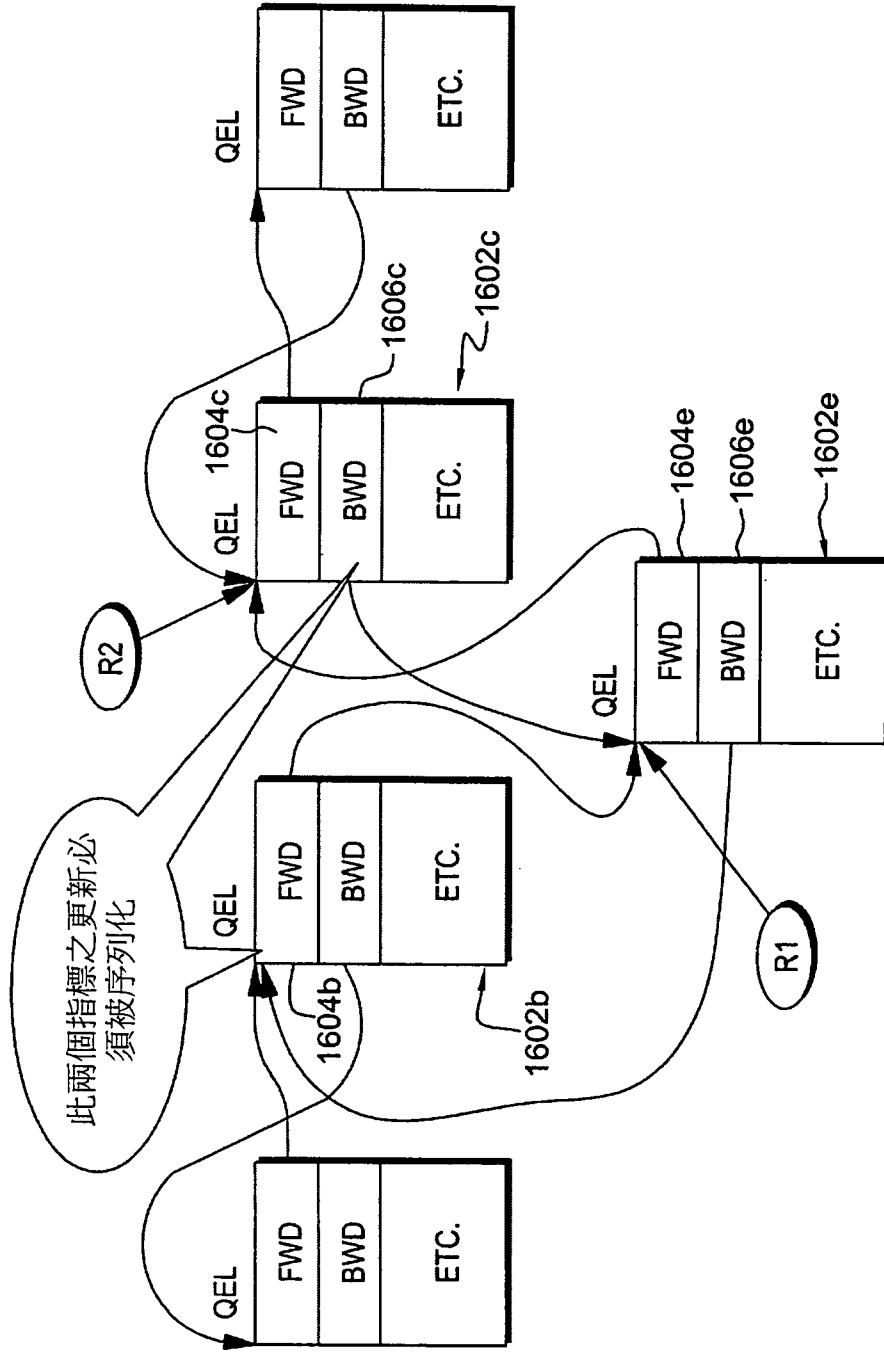


圖16B

電腦程式
產品
1700

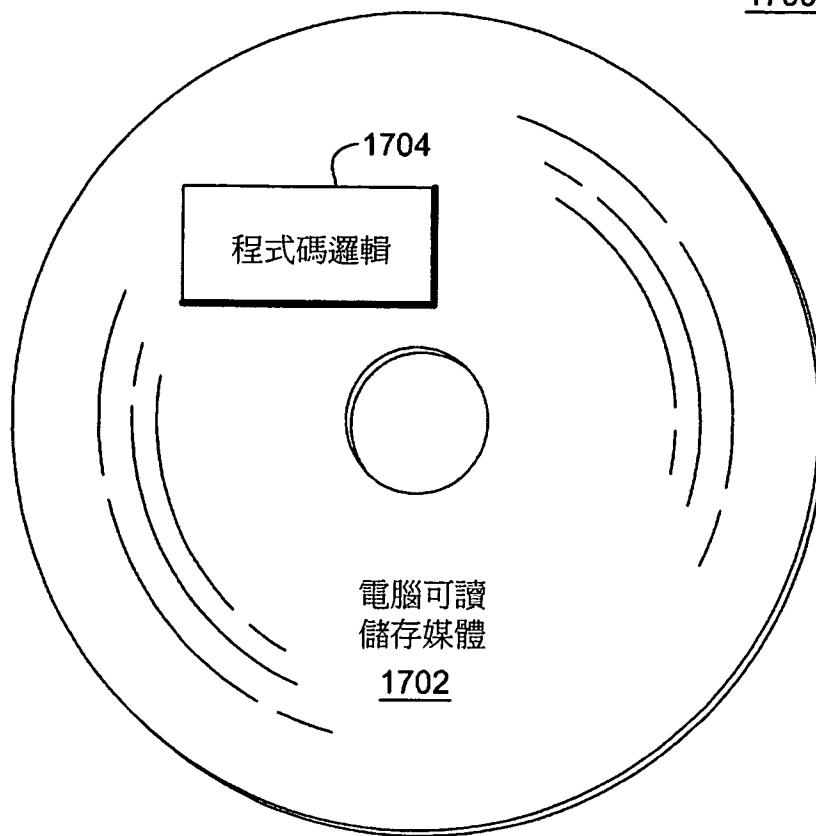


圖17

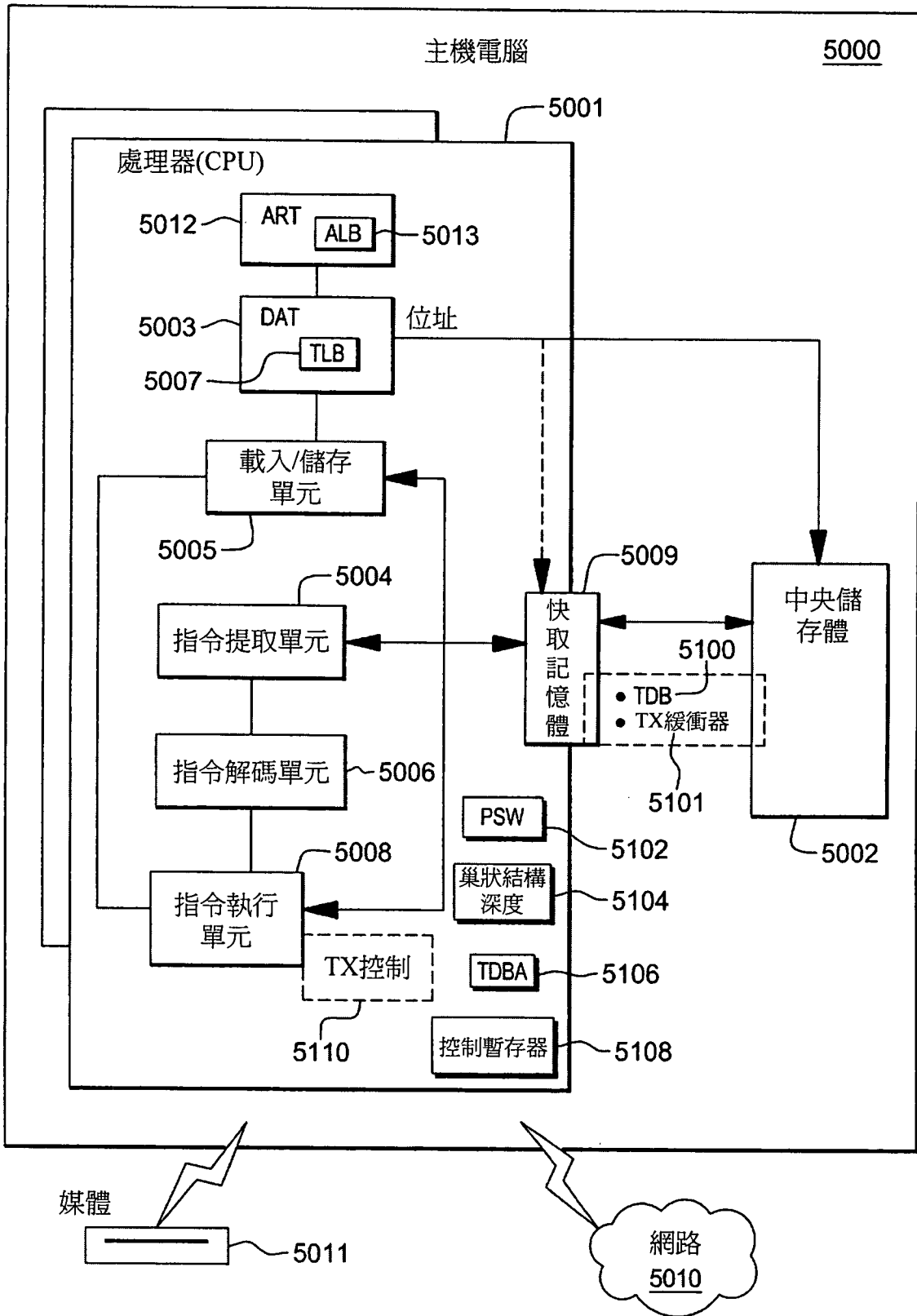


圖18

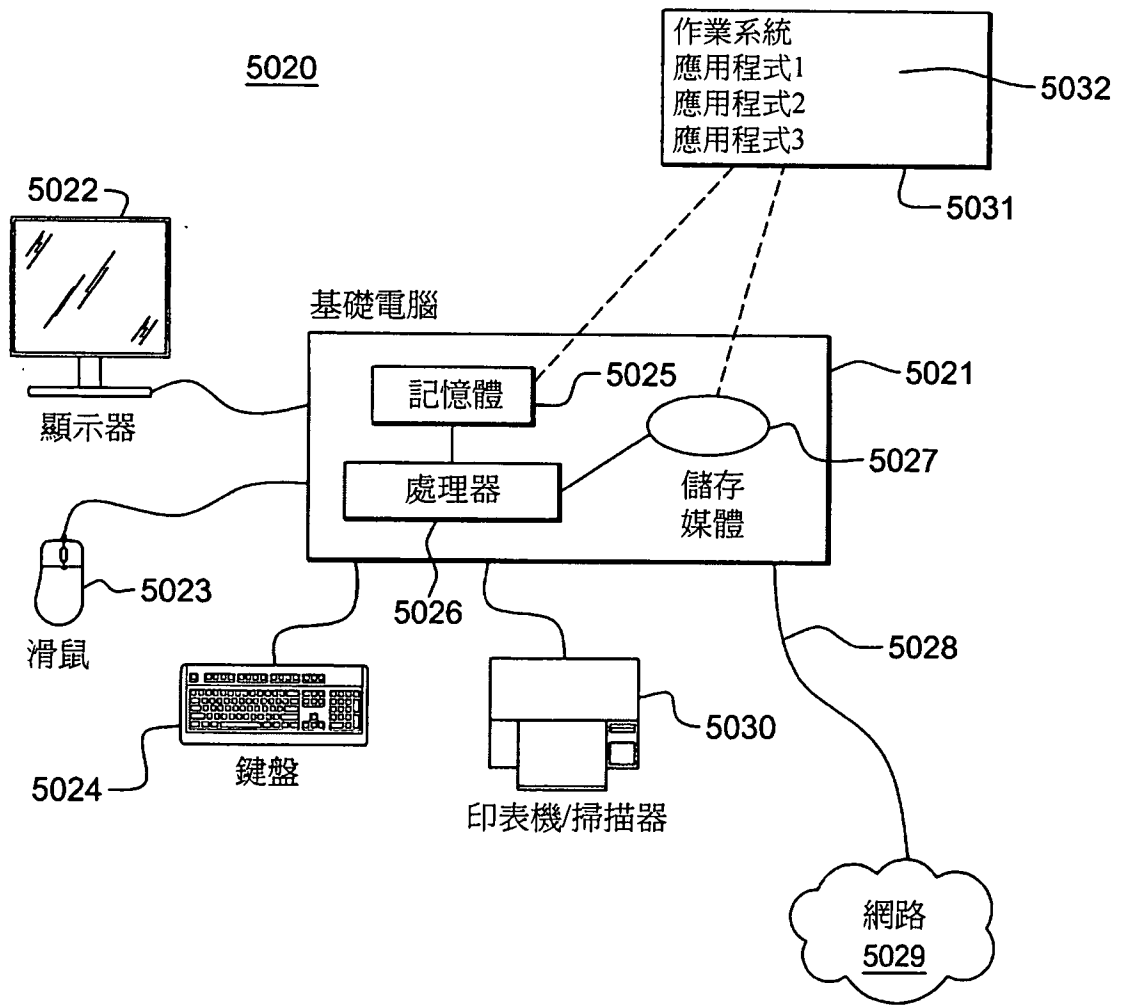


圖19

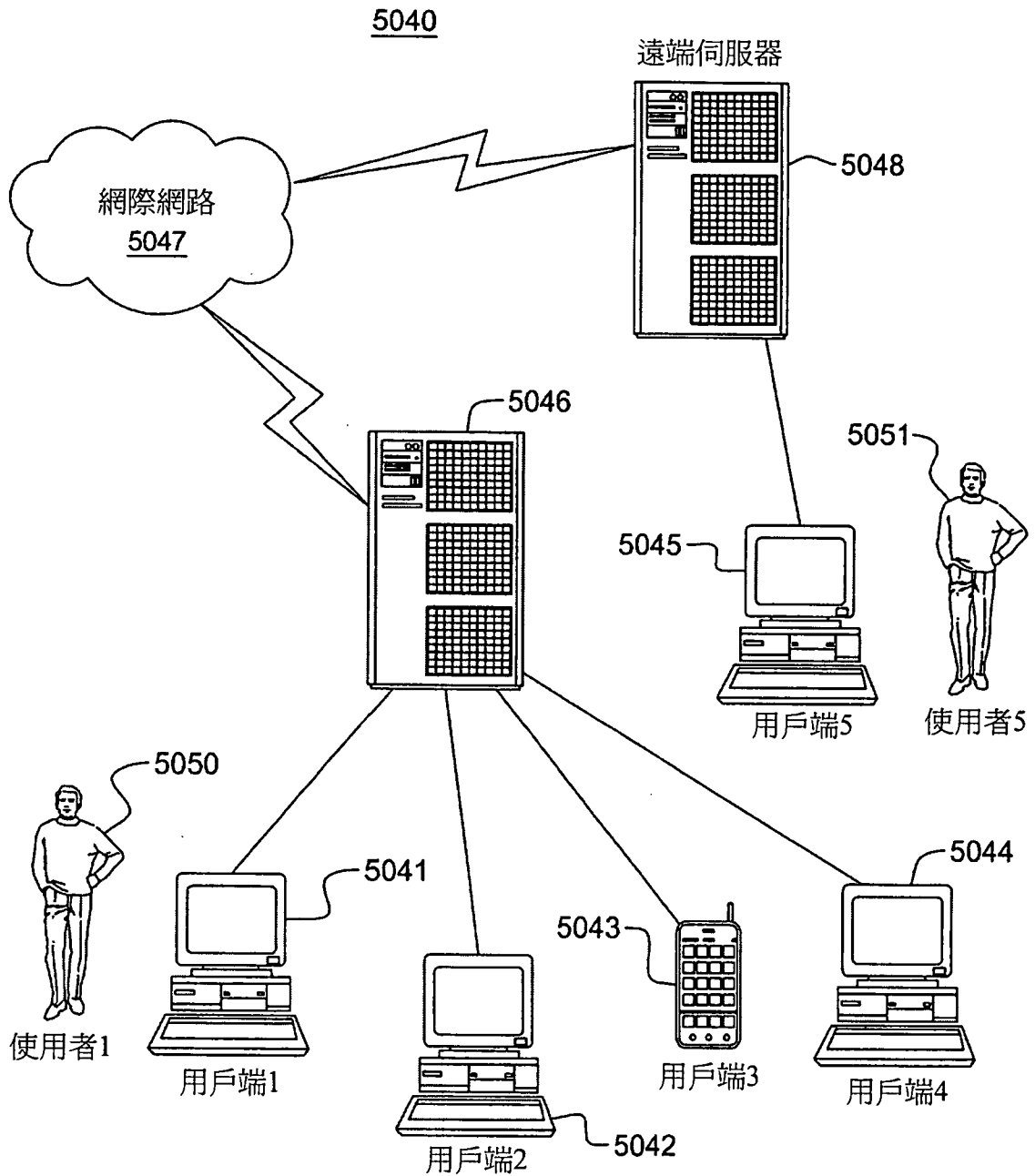


圖20

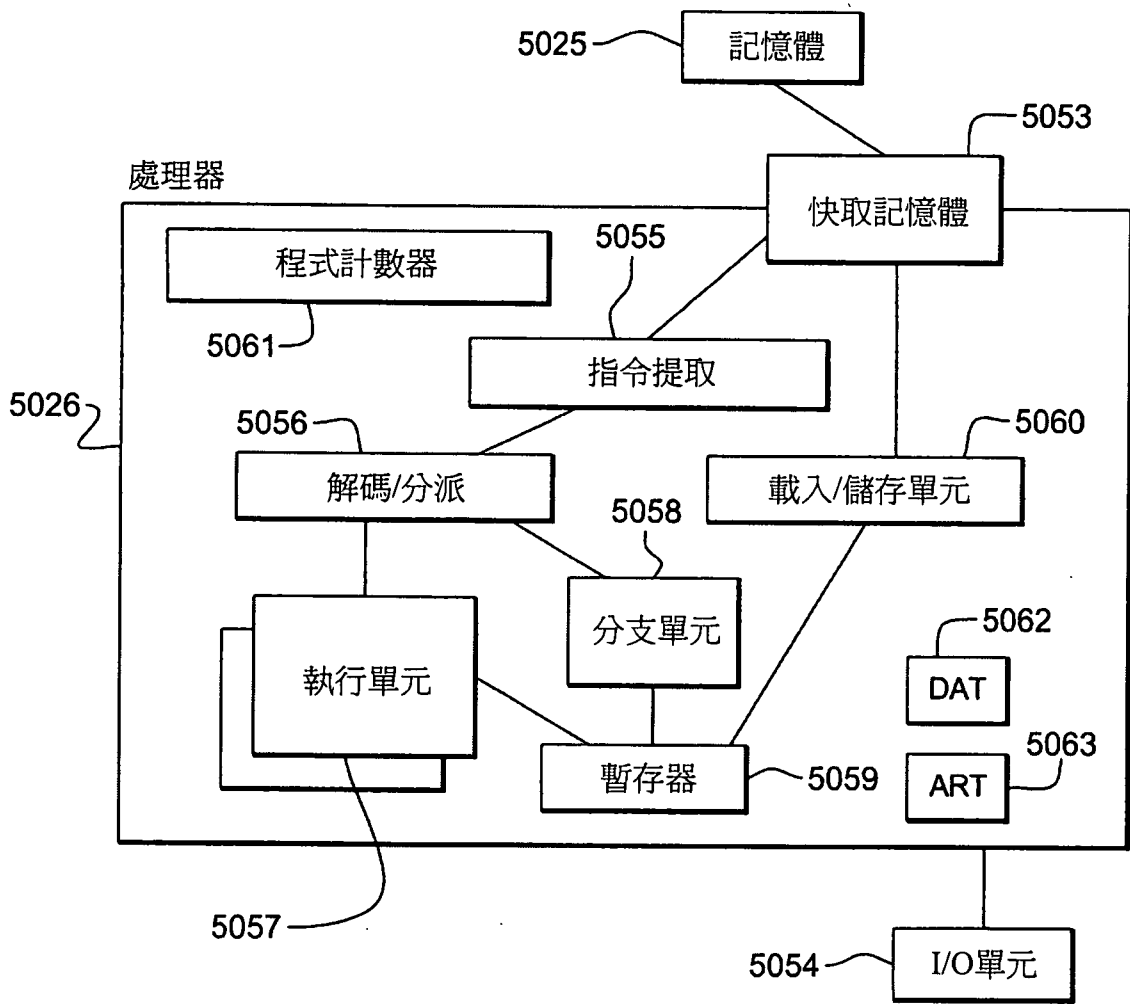


圖21

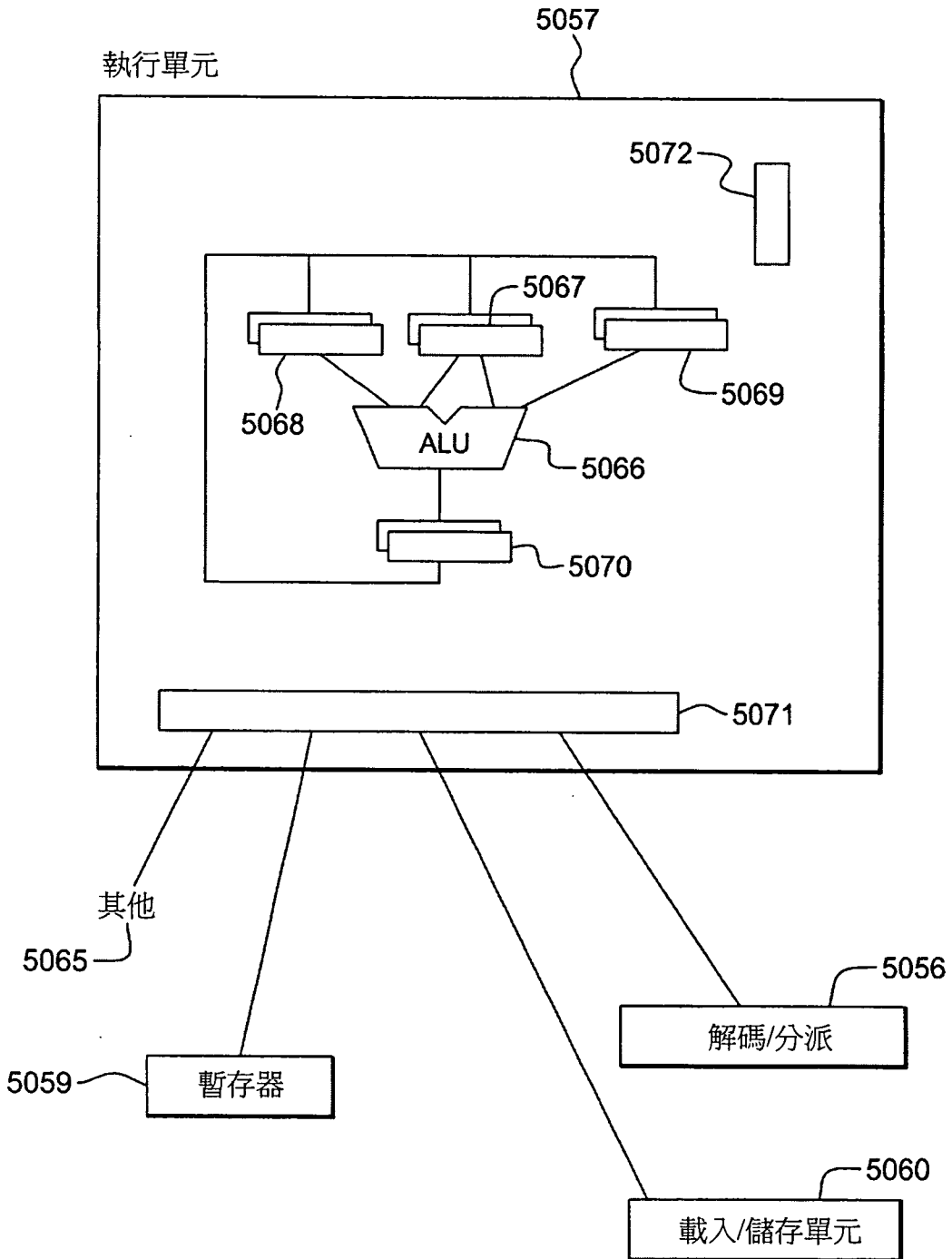


圖22A

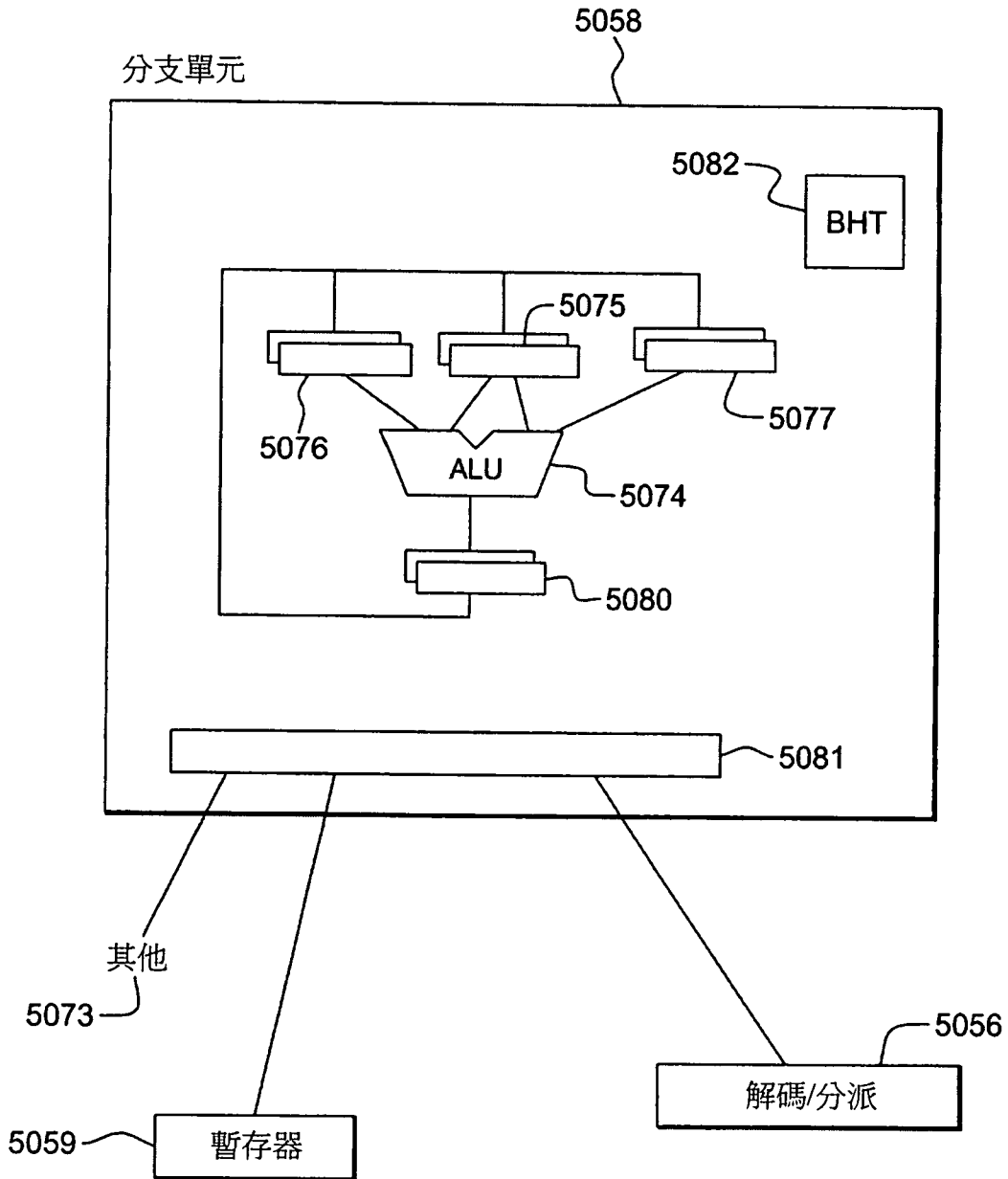


圖22B

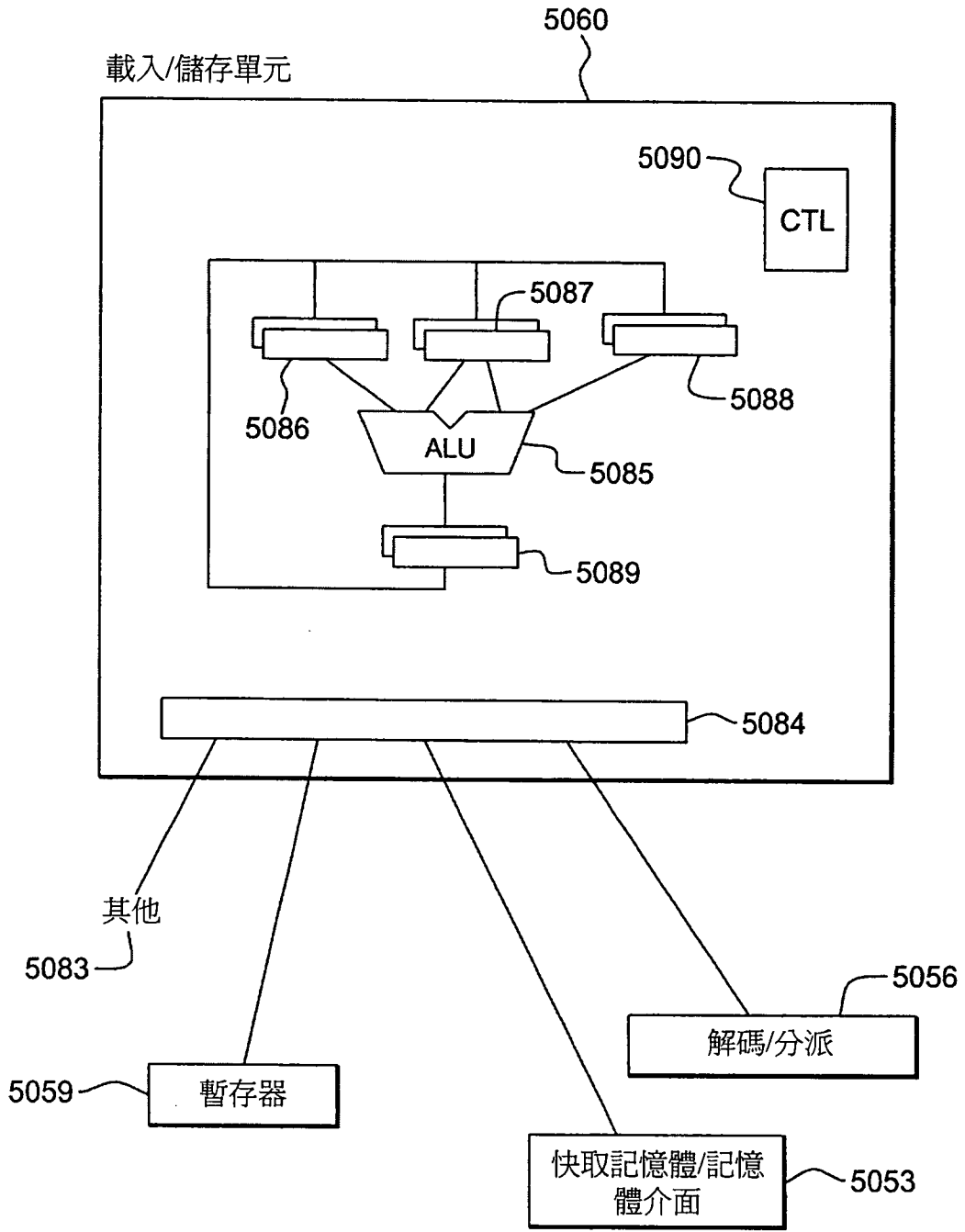


圖22C

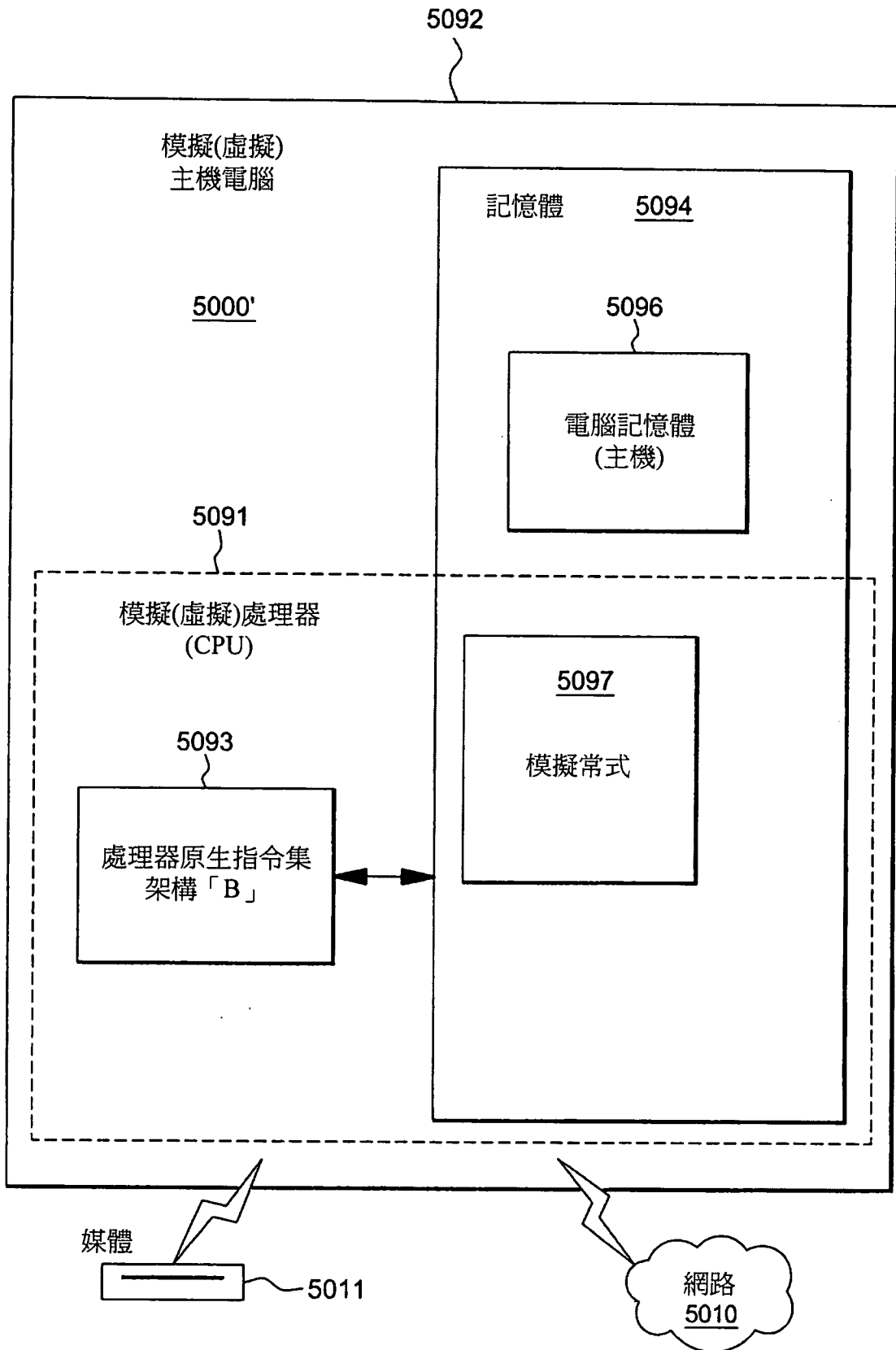


圖23